UNIVERSIDAD TECNOLÓGICA DE LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Tesis

Adaptación del modelo NLLB-200 para la traducción automática de la lengua quechua en la región de Cusco, 2026

Asesor:
Mag. Roldan Medina, Moises

Autor:
Aima Jalisto, Javier

Para optar el Título Profesional de:
Ingeniero de Sistemas e Informática

Cusco - Cusco - Perú
2026

Acta de sustentación

UNIVERSIDAD TECNOLÓGICA DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS E INFORMATICA

Acta N°: 04

ACTA DE SUSTENTACIÓN DE TÍTULO PROFESIONAL

En la ciudad de Cusco, a los 10 días del mes de julio del 2026, siendo las 09:00 horas, se reunieron los integrantes del Jurado designado por Resolución Sub Directoral N° 053-2026-UTEA-FC-FI-EPISeI-SD. de la Escuela Profesional de Ingeniería de Sistemas e Informática, Facultad de ingeniería:

Presidente	:	Mgt. Anaya Bellido, Tula
Dictaminante	:	Dra. Huaman Vargas, Jaqueline
Replicante	:	Dr. Mellado Pagan Juan, Pablo

Para evaluar la sustentación, en la modalidad de:

☒ Tesis ☐ Trabajo de suficiencia profesional

Titulada:
" Adaptación del modelo NLLB-200 para la traducción automática de la lengua quechua en la región de Cusco, 2026"

Desarrollado por el (los) Bachiller (es):

Br.: Aima Jalisto, Javier
(Apellidos y Nombres)

Br.: _____
(Apellidos y Nombres)

Para optar el Tituló Profesional de:
Ingeniero de Sistemas e Informática
(Denominación del Título)

Concluido el acto, el Jurado dictaminó que el (la) (los) mencionado(a) (s) bachiller (es) fue (ron) APROBADO(S) ☒ o DESAPROBADO(S) ☐:

Emitiéndose el calificativo final de:

Bachiller (Apellidos y Nombres)	Nota	Detalle (**)
Br. Aima Jalisto, Javier	12	APROBADO

Siendo las 11:50 horas concluyó la sesión, firmando los integrantes del Jurado.

Presidente: Mgt. Anaya Bellido, Tula
(Dr. Mg.). (Apellidos y Nombres)
(Firma)

Dictaminante: Dra. Huaman Vargas, Jaqueline
(Dr. Mg.). (Apellidos y Nombres)
(Firma)

Replicante: Dr. Mellado Pagan, Juan Pablo
(Dr. Mg.). (Apellidos y Nombres)
(Firma)

Nota: Desaprobado: 0-10; **Aprobado:** 11-20
(**): 0 a 10: Desaprobado, 11 a 15: Aprobado, 16 a 18: Aprobado Notable, 19 y 20: Aprobado con Distinción, Art. 18 RGGAT.

Reporte de similitud

11% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Coincidencias menores (menos de 10 palabras)

Fuentes principales

8% Fuentes de Internet
2% Publicaciones
9% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Metadatos

Datos del Autor		
Apellidos y nombres	:	Aima Jalisto, Javier
Tipo de Documento de Identidad	:	DNI
Número de Documento de Identidad	:	46545916
URL ORCID	:	https://orcid.org/0009-0001-4268-0230
Datos del Asesor		
Apellidos y nombres	:	Mag. Roldan Medina, Moises
Tipo de Documento de Identidad	:	DNI
Número de Documento de Identidad	:	44721071
URL ORCID	:	https://orcid.org/0009-0003-4649-5010
Datos de la Investigación		
Facultad	:	Facultad de Ingeniería
Escuela Profesional	:	Ingeniería de Sistemas e Informática
Línea de Investigación	:	Informática, Sociedad y Gestión del Conocimiento
Rango de años en que se realizó la investigación	:	Enero 2026 – junio 2026
Fuente de financiamiento	:	Autofinanciado
Porcentaje de similitud	:	11 %
URL de OCDE	:	https://purl.org/pe-repo/ocde/ford#2.02.04

Dedicatoria

A la memoria de mi padre, que aún guía mis pasos desde el recuerdo.

A mi madre y a mis hermanos, por ser la raíz de mis sueños y la fuerza que me permitió alcanzarlos.

Y a los guardianes de las lenguas originarias, que a través de los siglos preservaron la voz de nuestros ancestros frente al silencio del tiempo. Gracias a ellos, la memoria de un pueblo continúa viva en sus palabras.

Con gratitud y respeto, dedico este trabajo a todos ustedes.

Javier Aima Jalisto

Agradecimientos

Mi agradecimiento sincero a los docentes que contribuyeron a mi formación académica, por su dedicación, orientación y compromiso en la transmisión de conocimientos, los cuales han sido fundamentales para mi desarrollo profesional.

A mi asesor de tesis, por su orientación, disposición y valiosas recomendaciones durante el desarrollo de esta investigación, las cuales contribuyeron al logro de los objetivos planteados.

A mi familia, por su apoyo incondicional, paciencia y comprensión durante toda mi etapa universitaria, así como por brindarme la motivación necesaria para culminar esta etapa de mi formación.

A mis compañeros de clase, por el compañerismo, la colaboración y el apoyo mutuo durante el desarrollo de la carrera, lo cual enriqueció significativamente esta experiencia académica.

Finalmente, agradezco a la Escuela Profesional de Ingeniería de Sistemas e Informática, y a la Universidad Tecnológica de los Andes, por brindarme las herramientas, recursos y el entorno académico necesario para mi formación profesional.

Javier Aima Jalisto

Resumen

La traducción automática para lenguas de bajos recursos como el quechua enfrenta desafíos críticos debido a la escasez de datos digitales y la falta de modelos especializados, lo que limita la inclusión tecnológica de sus hablantes. La presente investigación tuvo como objetivo determinar el efecto de adaptación del modelo multilingüe NLLB-200 para optimizar la traducción automática de la lengua quechua collao. El estudio empleó un enfoque de investigación cuantitativo, con un tipo de investigación aplicada y un diseño preexperimental, utilizando la variante NLLB-200 de 600 millones de parámetros. Para lograr una adaptación eficiente en términos computacionales, se implementó el proceso de *fine-tuning* mediante la técnica LoRA, trabajando con un corpus paralelo final de 126,964 pares de oraciones tras un riguroso proceso de limpieza, normalización y filtrado. Los resultados obtenidos a través de diversos experimentos demostraron que el modelo con mejor desempeño alcanzó una puntuación de 9.50 en la métrica BLEU, 40.52 en ChrF y 35.81 en ChrF++, superando el desempeño del modelo base original que registró valores de 4.82, 32.99 y 28.20, respectivamente. Estos hallazgos evidencian que el uso de técnicas de entrenamiento eficiente permite duplicar la calidad de la traducción sin requerir una infraestructura computacional inalcanzable. Se concluye que la adaptación de modelos multilingües mediante la técnica LoRA constituye una estrategia viable para el desarrollo de tecnologías de traducción automática orientadas a lenguas de bajos recursos, favoreciendo su incorporación en entornos digitales.

Palabras clave: Traducción automática neuronal, quechua collao, NLLB-200, LoRA, *fine-tuning*, procesamiento de lenguaje natural.

Abstract

Machine translation for low-resource languages such as Quechua faces critical challenges due to the scarcity of digital data and the lack of specialized models, limiting the technological inclusion of its speakers. The objective of this study was to determine the effect of adapting the multilingual NLLB-200 model to optimize machine translation for the Collao variety of Quechua. The study adopted a quantitative research approach, with an applied research type and a pre-experimental design, using the NLLB-200 variant with 600 million parameters. To achieve computationally efficient adaptation, the fine-tuning process was implemented using the LoRA technique, employing a final parallel corpus of 126,964 sentence pairs obtained through a rigorous process of cleaning, normalization, and filtering. The experimental results showed that the model with the best performance achieved a BLEU score of 9.50, a ChrF score of 40.52, and a ChrF++ score of 35.81, outperforming the original base model, which obtained scores of 4.82, 32.99, and 28.20, respectively. These findings demonstrate that the use of efficient training techniques can double translation quality without requiring unattainable computational infrastructure. It is concluded that adapting multilingual models using the LoRA technique constitutes a viable strategy for developing machine translation technologies for low-resource languages, thereby promoting their integration into digital environments.

Keywords: Neural machine translation, Quechua Collao, NLLB-200, LoRA, fine-tuning, natural language processing.

Índice

Portada i

Acta de sustentación ii

Reporte de similitud iii

Metadatos iv

Dedicatoria v

Agradecimientos vi

Resumen vii

Abstract viii

Índice ix

Índice de tablas xi

Índice de figuras xii

Índice de anexos xiv

I. Introducción 15

II. Planteamiento del problema 17

2.1 Descripción y formulación del problema 17

2.2 Objetivos 19

2.2.1 Objetivo General 19

2.2.2 Objetivos Específicos 19

2.3 Justificación e importancia 20

2.4 Hipótesis 22

2.5 Variables 22

III. Marco Teórico 24

3.1 Antecedentes 24

3.2 Bases teóricas 28

3.3 Definición de términos 50

IV. Metodología 54
4.1 Tipo y nivel de investigación 54
4.2 Ámbito temporal y espacial 55
4.3 Población y muestra 55
4.4 Instrumentos 57
4.5 Procedimientos 58
4.6 Consideraciones éticas 75
V. Resultados y discusión 77
VI. Conclusiones 87
VII. Recomendaciones 89
VIII. Referencias 91
IX. Anexos 97

Índice de tablas

Tabla 1 Operacionalización de variables 23

Tabla 2 Variedades del quechua en el Perú 42

Tabla 3 Fuentes del corpus paralelo español-quechua collao 56

Tabla 4 Distribución del corpus 57

Tabla 5 Herramientas computacionales e instrumentos de medición utilizados en la investigación 58

Tabla 6 Ejemplo de estructura tabular del corpus paralelo 60

Tabla 7 Fuentes integradas en el corpus consolidado 61

Tabla 8 Ejemplos de normalización Unicode NFC 61

Tabla 9 Ejemplos de normalización de espacios y caracteres invisibles 62

Tabla 10 Ejemplos de registros eliminados durante la limpieza 63

Tabla 11 Ejemplos de separación de signos de puntuación 64

Tabla 12 Ejemplo ilustrativo de un par descartado por diferencia excesiva de longitud 65

Tabla 13 Configuración de recursos computacionales y parámetros de entrenamiento . 70

Tabla 14 Evolución de pérdida durante el entrenamiento 74

Tabla 15 Resultados de los grupos evaluados 77

Tabla 16 Tiempo de entrenamiento y uso de memoria GPU según la configuración evaluada 78

Tabla 17 Resumen de eficiencia computacional del mejor modelo 78

Tabla 18 Resultados del Grupo A: variación del tamaño del corpus 79

Tabla 19 Resultados del Grupo B: variación del parámetro r 80

Tabla 20 Resultados del Grupo C: variación del parámetro α 81

Tabla 21 Comparación entre el modelo base y el mejor modelo adaptado 82

Índice de figuras

Figura 1 Arquitectura del modelo Transformer 29

Figura 2 Cronología de los grandes modelos de lenguaje (LLMs) 31

Figura 3 Etapas del proceso de *fine-tuning* de los grandes modelos de lenguaje (LLMs) 34

Figura 4 Taxonomía de los métodos PEFT 36

Figura 5 Una comparación entre las actualizaciones de peso en el *fine-tuning* regular y el *fine-tuning* con LoRA 37

Figura 6 Flujo del procedimiento experimental 59

Figura 7 Estructuración tabular del corpus. 60

Figura 8 Algoritmo para la normalización Unicode NFC 62

Figura 9 Algoritmo para la normalización de espacios y caracteres invisibles en el corpus 62

Figura 10 Algoritmo para eliminar pares vacíos y pares con espacios 63

Figura 11 Algoritmo para la separación de signos de puntuación 64

Figura 12 Almacenamiento del corpus filtrado en formato JSONL. 66

Figura 13 Algoritmo para la división del corpus paralelo 67

Figura 14 Conversión de subconjuntos a formato Dataset de Hugging Face 67

Figura 15 Almacenamiento del conjunto de datos en formato Arrow 68

Figura 16 Configuración e inicialización del tokenizador NLLB-200. 69

Figura 17 Carga del modelo base NLLB-200 preentrenado 71

Figura 18 Configuración LoRA para el *fine-tuning* del modelo NLLB-200 71

Figura 19 Inserción de adaptadores LoRA sobre el modelo base 72

Figura 20 Configuración de hiperparámetros de entrenamiento 72

Figura 21 Configuración del DataCollator para los lotes de entrenamiento 73

Figura 22 Ejecución del entrenamiento mediante Seq2SeqTrainer 73

Figura 23 Generación de traducciones automáticas 74

Figura 24 Cálculo de métricas de evaluación automática (BLEU, ChrF y ChrF++) . . . 75
Figura 25 Comparación de configuraciones LoRA según ChrF++ 79
Figura 26 Efecto del tamaño del corpus en el desempeño del modelo 80
Figura 27 Impacto del parámetro r en el rendimiento del modelo. 81
Figura 28 Efecto del factor α en la calidad de traducción 82
Figura 29 Comparación del modelo base vs modelo adaptado 83

Índice de anexos

Anexo 1 Matriz de consistencia 98

Anexo 2 Comparación de traducciones entre el modelo base y el mejor modelo adaptado. 99

Anexo 3 Curvas de entrenamiento y convergencia del modelo adaptado 100

Anexo 4 Ficha técnica de configuración e hiperparámetros del Experimento 9 101

Anexo 5 Entorno de ejecución en Google Colab 102

Anexo 6 Entorno de ejecución en la plataforma RunPod. 103

Anexo 7 Galería de fotografías del desarrollo de la investigación. 104

I. Introducción

La traducción automática constituye una de las principales aplicaciones del procesamiento del lenguaje natural, permitiendo facilitar la comunicación entre personas que utilizan diferentes idiomas mediante modelos basados en inteligencia artificial. En los últimos años, el desarrollo de modelos multilingües ha ampliado significativamente la cobertura de idiomas y ha mejorado la calidad de las traducciones. Sin embargo, estos avances continúan siendo limitados para las lenguas de bajos recursos, como el quechua collao, debido a la escasa disponibilidad de corpus paralelos y recursos lingüísticos adecuados para su entrenamiento y adaptación.

En este contexto, la adaptación de modelos multilingües mediante técnicas de *fine-tuning* eficiente representa una alternativa para mejorar el desempeño de la traducción automática sin incrementar considerablemente los requerimientos computacionales. Entre estas técnicas destaca LoRA, la cual permite especializar modelos preentrenados entrenando únicamente una pequeña fracción de sus parámetros, favoreciendo un uso más eficiente de los recursos disponibles.

La presente investigación analiza el efecto de la adaptación del modelo multilingüe NLLB-200 mediante la técnica LoRA en la traducción automática de la lengua quechua collao. Para ello, se desarrolló la adaptación del modelo utilizando un corpus paralelo español–quechua collao, evaluando su desempeño mediante métricas automáticas de traducción.

La tesis se estructura en los siguientes capítulos. El Capítulo I presenta la introducción de la investigación. El Capítulo II desarrolla el planteamiento del problema, donde se describe la realidad problemática, se formulan los objetivos, la justificación, importancia,

hipótesis y variables del estudio. El Capítulo III presenta el marco teórico, incluyendo los antecedentes y las bases conceptuales relacionadas con la adaptación del modelo NLLB-200, la técnica LoRA y la traducción automática de la lengua quechua. El Capítulo IV describe la metodología empleada, detallando el tipo y nivel de investigación, el ámbito de estudio, la población, muestra, instrumentos y procedimientos aplicados para la adaptación y evaluación del modelo. El Capítulo V presenta los resultados obtenidos y la discusión de los hallazgos. Finalmente, los capítulos VI y VII contienen las conclusiones y recomendaciones de la investigación, respectivamente. Asimismo, se incluyen las referencias bibliográficas y anexos que complementan el desarrollo del estudio.

II. Planteamiento del problema

2.1 Descripción y formulación del problema

La traducción automática (TA) de la lengua quechua es de gran importancia en el Perú, pues se trata de la lengua originaria con mayor número de hablantes en el país. De acuerdo con el Censo Nacional de 2017, se identificaron 3 millones 799 mil 780 personas quechuahablantes, lo que representa un 13.6 % de la población total, existiendo además 5 millones 176 mil 809 personas que se autoidentifican como Quechua (INEI, 2017). Por ello, esta lengua constituye un medio esencial para preservar la identidad cultural y social de las comunidades andinas (Horna-Saldaña et al., 2025). Sin embargo, el desarrollo de las tecnologías lingüísticas impulsadas por la inteligencia artificial (IA) aún presenta limitaciones importantes, principalmente debido a la escasa disponibilidad de datos de evaluación y a la falta de modelos especializados para lenguas de bajos recursos (Ebrahimi et al., 2024) como el quechua. En este contexto, el *fine-tuning* del modelo *No Language Left Behind* (NLLB-200) mediante técnicas como *Low-Rank Adaptation* (LoRA) se presenta como una alternativa clave para optimizar la traducción al quechua (Garcia Gilabert et al., 2024).

La problemática se manifiesta principalmente en regiones del país donde se concentra la mayor población quechuahablante, particularmente en los Andes sureños (Huancavelica, Ayacucho, Apurímac, Cuzco y Puno), además de la capital de Lima (INEI, 2017) y donde el acceso a tecnologías de traducción es aún limitado. Uno de los principales factores que influyen en esta situación es que las lenguas nativas de bajos recursos, enfrentan una grave desventaja debido a la escasez de corpus paralelos para desarrollar modelos de TA (Tafa et al.,

2025). Además, los modelos multilingües suelen mostrar una prioridad hacia las lenguas con altos recursos como el inglés, francés, chino, ruso o español, lo que deja en desventaja a las lenguas de bajos recursos como el quechua collao (Haddow et al., 2022; Mager et al., 2023).

Las consecuencias de esta situación se evidencian en que los hablantes de lenguas nativas como el quechua collao se ven forzados a traducir hacia una lengua dominante por la demanda del mercado (Guzman & Avalos, 2025), lo que contribuye a ampliar la brecha tecnológica que enfrenta esta lengua y limita su incorporación en los entornos digitales. Frente a esta realidad, los grandes modelos de lenguaje (LLMs) presentan un potencial significativo, ya que pueden adaptarse a tareas específicas como la TA mediante procesos de *fine-tuning* (Santos et al., 2025). En este contexto, se hace necesario investigar estrategias para adaptar el modelo NLLB-200 (Costa-jussà et al., 2022) a la TA de la lengua quechua mediante técnicas de adaptación eficiente como LoRA (E. Hu et al., 2021). De esta forma, la investigación busca aportar al desarrollo de tecnologías de traducción automática para lenguas de bajos recursos, contribuyendo al desarrollo de herramientas tecnológicas que permitan una mayor presencia digital de la lengua quechua.

Asimismo, esta problemática se relaciona con los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas, especialmente con el ODS 10, orientado a la reducción de las desigualdades, al abordar las limitaciones tecnológicas que enfrentan las lenguas originarias en entornos digitales. De igual manera, se vincula con el ODS 9, relacionado con la innovación e infraestructura, mediante la adaptación de tecnologías basadas en IA para la TA de lenguas de bajos recursos como el quechua.

2.1.1 Problema General

¿Cuál es el efecto de la adaptación del modelo NLLB-200 en la traducción automática de la lengua quechua en la región de Cusco, 2026?

2.1.2 Problemas Específicos

- ¿En qué medida la aplicación de la técnica LoRA permite adaptar el modelo NLLB-200 para la traducción automática de la lengua quechua en la región de Cusco, 2026?
- ¿Cuál es el rendimiento del modelo NLLB-200 adaptado en la traducción automática de la lengua quechua en la región de Cusco, 2026?
- ¿En qué medida el modelo NLLB-200 adaptado mejora el desempeño respecto al modelo base original en la traducción automática de la lengua quechua en la región de Cusco, 2026?

2.2 Objetivos

2.2.1 Objetivo General

Determinar el efecto de la adaptación del modelo NLLB-200 en la traducción automática de la lengua quechua en la región de Cusco, 2026.

2.2.2 Objetivos Específicos

- Aplicar la técnica LoRA para adaptar el modelo NLLB-200 en la traducción automática de la lengua quechua en la región de Cusco, 2026.
- Evaluar el rendimiento del modelo NLLB-200 adaptado mediante métricas en la traducción automática de la lengua quechua en la región de Cusco, 2026.
- Comparar el desempeño del modelo NLLB-200 frente al modelo base en la traducción automática de la lengua quechua en la región de Cusco, 2026.

2.3 Justificación e importancia

2.3.1 Justificación

Desde un enfoque teórico, esta investigación se fundamenta en la necesidad de cerrar la brecha de rendimiento entre las lenguas de altos recursos y las de bajos recursos en el ámbito de la TA. Costa-jussà et al. (2022) explican que, aunque los modelos multilingües permiten el aprendizaje entre idiomas relacionados, la representación de cientos de lenguas con alta calidad sigue siendo un desafío abierto. En este contexto, el estudio aporta valor al analizar cómo la técnica LoRA permite adaptar un LLM de manera eficiente. Según E. Hu et al. (2021), esta técnica optimiza solo una fracción mínima de los parámetros del modelo original. Así, la tesis busca llenar un vacío de conocimiento sobre la aplicación conjunta de NLLB-200 y LoRA en la TA entre el español y el quechua collao.

En términos prácticos, esta investigación busca desarrollar un sistema de TA que facilite la comunicación entre hablantes de quechua y español. Costa-jussà et al. (2022) señalan que la falta de traducciones de calidad es una barrera para la educación y la movilidad social, ya que gran parte del contenido de internet sigue priorizando idiomas dominantes como el inglés. Al desarrollar un sistema de traducción para el quechua, se propone una innovación que puede mejorar la calidad de vida de sus hablantes. Como se menciona en los objetivos de la conferencia LT4All 2025, estas tecnologías lingüísticas son fundamentales para democratizar la inteligencia artificial y reducir las desigualdades (UNESCO, 2025).

Desde el ámbito metodológico, esta investigación tiene valor porque propone una forma organizada y replicable de adaptar el modelo NLLB-200 para la TA entre el español y el quechua collao mediante la técnica LoRA. Hernández-Sampieri y Mendoza Torres (2023) indican que una investigación se justifica metodológicamente cuando puede contribuir a crear nuevos métodos, instrumentos o técnicas de investigación que sirvan de base para estudios

posteriores. En ese sentido, el proceso de preparación de datos, entrenamiento, evaluación constituye una referencia metodológica para futuras investigaciones.

Desde la perspectiva social, esta investigación aporta al desarrollo de tecnologías orientadas al procesamiento automático del quechua collao, una lengua considerada de bajos recursos en el ámbito digital. Según la UNESCO (2025), las lenguas son un medio para la transmisión del conocimiento, la cultura y la identidad, por lo que el fortalecimiento de tecnologías multilingües favorece una mayor inclusión en los entornos digitales. En ese sentido, la adaptación del modelo NLLB-200 representa un aporte para futuras aplicaciones de TA en quechua collao y contribuye a ampliar las alternativas tecnológicas disponibles para esta lengua.

2.3.2 Importancia

La importancia de la presente investigación radica en que permite determinar el efecto de la adaptación del modelo NLLB-200 mediante la técnica LoRA en la traducción automática de la lengua quechua, generando evidencia sobre la aplicación de técnicas de *fine-tuning* eficiente en lenguas de bajos recursos. Asimismo, sus resultados aportan una alternativa para el desarrollo de sistemas de traducción automática con menores requerimientos computacionales, lo que facilita su implementación en contextos con recursos limitados. Finalmente, la investigación contribuye al desarrollo de recursos tecnológicos para la lengua quechua, favoreciendo su incorporación en los entornos digitales y apoyando iniciativas orientadas a la preservación y difusión de su patrimonio lingüístico y cultural.

2.4 Hipótesis

2.4.1 Hipótesis General

La adaptación del modelo NLLB-200 mejora la calidad de la traducción automática de la lengua quechua en la región de Cusco, 2026.

2.4.2 Hipótesis Específicas

- La aplicación de la técnica LoRA en el modelo NLLB-200 permite su adaptación efectiva en la traducción automática de la lengua quechua en la región de Cusco, 2026.
- El modelo NLLB-200 adaptado alcanza métricas consistentes en la traducción automática de la lengua quechua en la región de Cusco, 2026.
- El modelo NLLB-200 adaptado obtiene puntuaciones superiores al modelo base en la traducción automática de la lengua quechua en la región de Cusco, 2026.

2.5 Variables

Tabla 1
Operacionalización de variables

Adaptación del modelo NLLB-200 para la traducción automática de la lengua quechua en la región de Cusco, 2026				
Variables	**Dimensiones**	**Indicadores**	**Descripción**	**Escala de medición**
Variable independiente: *Adaptación del modelo NLLB-200* Es cuando un modelo multilingüe preentrenado puede ser adaptado a tareas específicas mediante *fine-tuning*, lo que implica entrenarlo adicionalmente con un conjunto de datos particular, permitiendo mejorar su desempeño con recursos computacionales limitados (Costa-jussà et al., 2022; Parthasarathy et al., 2024).	**Dimensión 1:** *Cantidad de corpus paralelo* La cantidad de corpus paralelos se refiere al volumen del corpus (colección de textos) que consiste en pares de oraciones o segmentos alineados en dos idiomas diferentes (Do Campo Bayón, 2023).	1.1. Tamaño del corpus	Número total de pares de oraciones español-quechua durante el proceso de *fine-tuning*.	Cuantitativa discreta. n° de pares de oraciones.
	Dimensión 2: *Hiperparámetros de LoRA* Los hiperparámetros de LoRA son configuraciones técnicas que controlan el grado de adaptación del método Low-Rank Adaptation durante el *fine-tuning* de LLMs, afectando su eficiencia y rendimiento (E. Hu et al., 2021).	2.1. Rango (r)	Número de dimensiones latentes usadas para descomponer las matrices de pesos del modelo.	Cuantitativa discreta. Valores esperados: $r \in \{8, 16, 32\}$.
		2.2. Factor de escala (α)	Parámetro que amplifica la contribución de las matrices adaptadas respecto al modelo base.	Cuantitativa discreta. Valores esperados: $\alpha \in \{8, 16, 32\}$.
Variable dependiente: *Traducción automática de la lengua quechua* La traducción automática (TA) es un proceso computacional que convierte un texto de un idioma a otro de forma autónoma, es decir, sin intervención humana (Silva et al., 2024).	**Dimensión 1:** *Calidad de la traducción* La calidad de la traducción es un concepto relativo y complejo que se refiere tanto al producto final (es decir, al material traducido) como al proceso. La evaluación de la calidad en la TA se realiza mediante un enfoque pragmático que combina la evaluación humana y métricas automáticas (Do Campo Bayón, 2023).	1.1. BLEU	Puntuación que mide la similitud entre la traducción generada y una traducción de referencia humana.	Cuantitativa continua. Rango: 0–100.
		1.2. ChrF++	Métrica basada en coincidencia de caracteres y n-gramas para evaluar similitud entre traducciones.	Cuantitativa continua. Rango: 0–100.
	Dimensión 2: *Eficiencia computacional* La eficiencia computacional en el contexto de la TA, se define como la capacidad de los modelos para lograr bajo costo y alta eficiencia en sus operaciones (H. Wang et al., 2022).	2.1. Tiempo total de entrenamiento	Duración del proceso de *fine-tuning*.	Cuantitativa continua. segundos/minutos/horas
		2.2. Uso de memoria GPU	Memoria promedio utilizada por la GPU durante el entrenamiento.	Cuantitativa continua. GB

III. Marco Teórico

3.1 Antecedentes

Hus et al. (2025) presentaron su investigación titulada “Machine Translation Using Grammar Materials for LLM Post-Correction”, desarrollada en George Mason University, Virginia – Estados Unidos. El estudio tuvo como objetivo principal optimizar los sistemas de traducción automática para lenguas indígenas mediante una estrategia de poscorrección basada en grandes modelos de lenguaje (LLM), empleando materiales lingüísticos de referencia para refinar las salidas de un traductor previo. Para ello, aplicaron una metodología experimental con un enfoque híbrido, utilizando el modelo GPT-4o-mini para editar y mejorar las propuestas de traducción generadas por modelos NLLB 3.3B ajustados específicamente para cada par de lenguas. En cuanto a los resultados obtenidos para la dirección de español a quechua (es→quy), el sistema de poscorrección con LLM alcanzó puntuaciones de 3.8 en BLEU, 36.3 en ChrF y 31.68 en ChrF++, mientras que el modelo NLLB ajustado obtuvo 3.7 en BLEU, 38.02 en ChrF y 32.7 en ChrF++ en la misma dirección. La investigación concluyó que, para la traducción hacia el quechua, el sistema NLLB resulta más efectivo y computacionalmente eficiente, determinando que, si bien los LLM son útiles para comprender la estructura de las lenguas indígenas, los modelos dedicados siguen siendo superiores para generar texto con precisión en estas lenguas.

Garcia Gilabert et al. (2024) desarrollaron la investigación titulada “BSC Submission to the AmericasNLP 2024 Shared Task”, realizada en el Barcelona Super Computing Center, Madrid – España. El estudio tuvo como propósito desarrollar sistemas de traducción automá-

tica del español al quechua y del español al guaraní para la tarea compartida de AmericasNLP 2024. Su metodología contempló el *fine-tuning* de diversos modelos NLLB-200 mediante la técnica de adaptadores LoRA, experimentando con datos multilingües y sintéticos. El sistema desarrollado logró una puntuación ChrF++ de 38.21 para la dirección español–quechua (es→qu), superando a los modelos base de referencia de la competencia AmericasNLP. Los autores concluyeron que el incremento del tamaño del modelo no garantiza un mejor rendimiento ante datos escasos, que la integración de datos sintéticos y multilingües favorece la calidad de traducción al quechua, y que el ajuste con LoRA alcanza un rendimiento comparable al entrenamiento completo actualizando únicamente el 14.2 % de los parámetros totales.

DeGenaro y Lupicki (2024) llevaron a cabo la investigación titulada “Experiments in Mamba Sequence Modeling and NLLB-200 Fine-Tuning for Low Resource Multilingual Machine Translation”, presentada en la Association for Computational Linguistics, México. El trabajo se propuso desarrollar sistemas de traducción automática del español hacia once lenguas indígenas. Para ello, evaluaron dos enfoques multilingües: el *fine-tuning* de la variante destilada de 600M parámetros de NLLB-200 bajo diversas configuraciones y el entrenamiento desde cero de una arquitectura basada en Mamba State Space Modeling. El modelo NLLB-200 adaptado alcanzó un ChrF++ de 36.02 en la dirección español–quechua, superando la línea base de AmericasNLP, mientras que la arquitectura Mamba presentó un rendimiento menor en dicho escenario. El estudio concluyó que la adaptación de NLLB-200 resulta altamente efectiva para la traducción en lenguas indígenas de bajos recursos.

Estos estudios coinciden en que NLLB-200 constituye una alternativa sólida para la traducción de lenguas indígenas y que las técnicas de *fine-tuning* permiten mejorar su desempeño.

Liang et al. (2025) presentaron el trabajo titulado “Toward Low-Resource Languages Machine Translation: A Language-Specific Fine-Tuning With LoRA for Specialized Large Language Models”, desarrollado conjuntamente entre la Universiti Tunku Abdul Rahman

(Malasia) y la Guangxi Minzu University (China). La investigación se orientó a mejorar la traducción automática en lenguas con recursos limitados mediante el método LSFTL (*Language-Specific Fine-Tuning with LoRA*), optimizando las capas de atención y redes *feed-forward* del Transformer. Su propuesta consistió en un ajuste específico por idioma que preserva la mayor parte de los parámetros originales del modelo, manteniendo la estabilidad y la calidad de salida. Los resultados mostraron incrementos de 1 a 3 puntos en métricas COMET para lenguas asiáticas de bajos recursos no centradas en inglés. La investigación concluyó que este enfoque representa una alternativa eficiente y robusta frente al entrenamiento completo, señalando asimismo que tiene limitaciones en idiomas de recursos extremadamente bajos, con datos paralelos inferiores a 10 000 pares de oraciones.

Chen et al. (2024) desarrollaron la investigación titulada "QueEn: A Large Language Model for Quechua-English Translation", realizada en The University of Georgia, Estados Unidos. El estudio tuvo como objetivo principal implementar un modelo especializado para la traducción quechua–inglés. Para ello, emplearon técnicas de *Retrieval-Augmented Generation* (RAG) combinadas con *Parameter-Efficient Fine-Tuning* mediante LoRA. El modelo QueEn alcanzó una puntuación BLEU de 17.6 frente a 1.5 registrado por modelos GPT estándar. El estudio concluyó que la integración de RAG y técnicas de ajuste eficiente constituye una estrategia prometedora para potenciar la traducción de lenguas de bajos recursos manteniendo un bajo costo computacional.

Por otro lado, investigaciones más recientes centran su atención en la eficiencia del *fine-tuning*, proponiendo estrategias como LoRA y LSFTL para reducir el costo computacional sin afectar la calidad de traducción.

Gow-Smith y Sánchez Villegas (2023) presentaron su investigación titulada "Sheffield's Submission to the AmericasNLP Shared Task on Machine Translation into Indigenous Languages", desarrollada en la Universidad de Sheffield, Reino Unido. El estudio tuvo como objetivo desarrollar un sistema de traducción automática multilingüe del español a once

lenguas indígenas a partir de NLLB-200. Aplicaron diversas variantes del modelo y técnicas de ensamblado (*ensembling*) para optimizar el rendimiento, logrando una puntuación ChrF de 39.52 en la dirección español–quechua. Los autores concluyeron que el entrenamiento multilingüe y el ensamblado de variantes basadas en NLLB-200 constituyen una estrategia sumamente eficaz para mejorar la TA en lenguas de bajos recursos.

Ahmed et al. (2023) llevaron a cabo el estudio titulado "Enhancing Spanish-Quechua Machine Translation with Pre-Trained Models and Diverse Data Sources: LCT-EHU at AmericasNLP Shared Task", realizado en la Universidad del País Vasco, España. La investigación tuvo como propósito mejorar la traducción español–quechua mediante el uso de modelos existentes en diferentes pares de idiomas. Emplearon estrategias de *fine-tuning* sobre modelos multilingües (español–finés y español–inglés), retrotraducción y corpus paralelos sintéticos. Como resultado, el modelo basado en español–finés alcanzó un rendimiento de hasta 38.59 en la métrica ChrF, superando otras configuraciones. La investigación concluyó que el aprovechamiento de modelos preentrenados mejora la traducción automática en lenguas de bajos recursos.

En conjunto, los antecedentes revisados muestran que las investigaciones recientes se han enfocado en mejorar la TA para lenguas de bajos recursos mediante el uso de LLMs, técnicas de *fine-tuning* como LoRA y diferentes estrategias para aprovechar mejor los datos disponibles. En la mayoría de los estudios se observa que modelos como NLLB-200 logran un mejor desempeño que otros enfoques cuando se trabaja con recursos lingüísticos limitados. Sin embargo, gran parte de estas investigaciones se desarrollaron en el contexto de competencias como AmericasNLP o en distintos pares de idiomas, centrándose principalmente en mejorar las métricas de traducción. Por ello, aún existe la necesidad de evaluar este tipo de modelos en escenarios específicos como la traducción español–quechua collao, considerando no solo la calidad de la traducción mediante las métricas BLEU y ChrF++, sino también la eficiencia computacional alcanzada durante el proceso de adaptación utilizando la técnica

LoRA.

3.2 Bases teóricas

3.2.1 Adaptación del Modelo NLLB-200

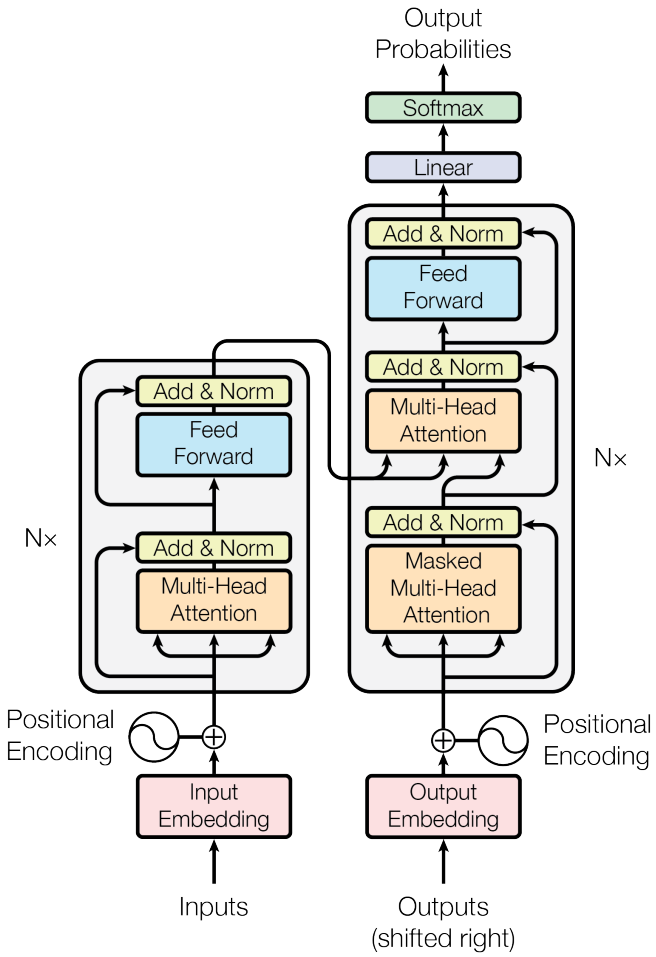
3.2.1.1 Arquitectura Transformer

La arquitectura Transformer es un modelo de red neuronal propuesto por Vaswani et al. (2023) que revolucionó el campo de la traducción automática neuronal (TAN) y el procesamiento del lenguaje natural (PLN). El Transformer se basa únicamente en mecanismos de atención (*attention mechanisms*) y fue diseñado para prescindir por completo de la recurrencia y las convoluciones, lo que representó un cambio radical respecto a los modelos anteriores, como las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN) (Vaswani et al., 2023).

Autores posteriores como Liu et al. (2025) señalan que este modelo es un sistema de aprendizaje profundo diseñado específicamente para procesar datos secuenciales, destacando su superioridad en paralelización y rendimiento sobre las redes neuronales populares del pasado.

Para este estudio, el Transformer es fundamental al ser la base tecnológica del modelo NLLB-200, permitiendo que el sistema procese de manera paralela las secuencias del quechua y capture relaciones lingüísticas complejas a pesar de la distancia entre palabras.

Figura 1
Arquitectura del modelo Transformer



Nota. Reproducido de Vaswani et al. (2023).

La arquitectura del Transformer está conformada por dos módulos principales, el codificador (*encoder*) y el decodificador (*decoder*), los cuales se apilan repetidamente. En esta estructura, el modelo emplea mecanismos de autoatención y capas completamente conectadas aplicadas de forma puntual en ambos componentes, como se observa en las mitades izquierda y derecha de la Figura 1, respectivamente (Vaswani et al., 2023).

3.2.1.1.1 Codificador (encoder)

Recibe la secuencia de entrada (el texto fuente) y la mapea a una sucesión de representaciones continuas, está compuesto por una pila de 6 capas idénticas (Vaswani et al., 2023).

Do Campo Bayón (2023) explican que su función principal es transformar la secuen-

cia de entrada (idioma origen) en una representación continua de alta dimensionalidad, es decir, transforma la frase original en una lista de vectores a los que asigna un símbolo.

El codificador es un componente fundamental, ya que se encarga de procesar la oración de entrada en español y generar representaciones internas que contienen la información necesaria para que el decodificador produzca la traducción al quechua collao, la lengua meta.

3.2.1.1.2 Decodificador (decoder)

Genera la secuencia de salida (la traducción) elemento por elemento, utilizando los símbolos previamente generados como entrada adicional para predecir el siguiente token, tanto el codificador como el decodificador están compuestos por una pila de 6 capas idénticas (Vaswani et al., 2023).

Según Liu et al. (2025), el decodificador incluye capas de atención adicionales que utilizan las representaciones del codificador y aplica máscaras para evitar el acceso a información futura durante la generación de texto, asegurando que cada predicción se base únicamente en el contexto previo.

En este estudio, el decodificador es el encargado de producir el texto en quechua, manejando la compleja morfología aglutinante y sintáctica de esta lengua a partir de las representaciones aprendidas.

3.2.1.2 Modelos de lenguaje preentrenados

Los modelos de lenguaje preentrenados (PLMs) son sistemas de IA que utilizan arquitecturas de aprendizaje profundo, principalmente basadas en Transformer, se entrenan en grandes volúmenes de datos de texto no etiquetado a escala web (Minaee et al., 2025).

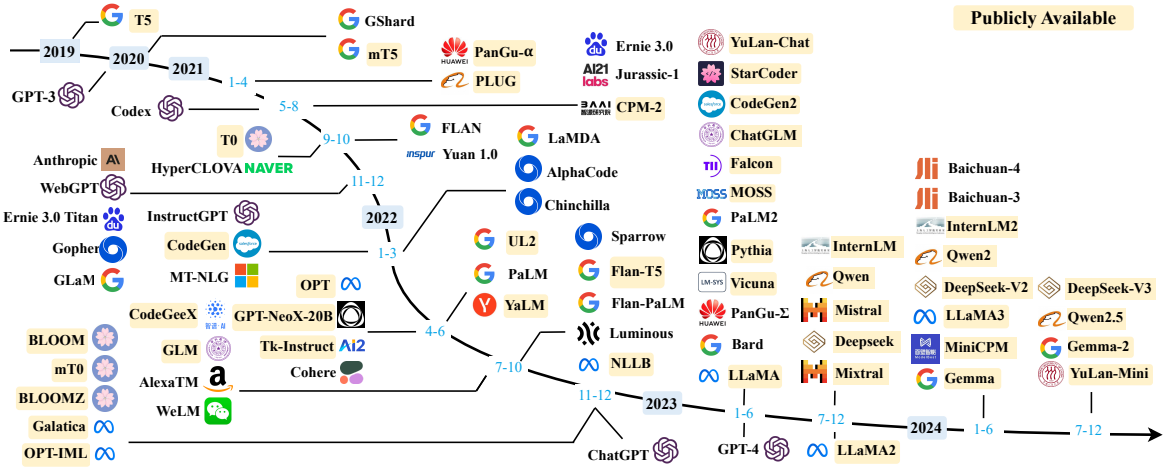
La última generación de PLMs son los grandes modelos de lenguaje (LLMs), estos se definen principalmente por su escala sin precedentes, conteniendo típicamente decenas a cientos de miles de millones de parámetros como GPT-4, PaLM y LLaMA; esta escala se hizo

posible gracias a la disponibilidad de grandes cantidades de datos y avances en la capacidad de cómputo (Minaee et al., 2025; Naveed et al., 2024).

Minaee et al. (2025) indican que estos modelos sirven como una base de conocimiento que puede ser adaptada para diversas tareas posteriores, como la TA o el análisis de sentimientos.

Figura 2

Cronología de los grandes modelos de lenguaje (LLMs)



Nota. Reproducido de Zhao et al. (2025). La figura presenta la cronología de los grandes modelos de lenguaje (LLMs) con más de 10 mil millones de parámetros desarrollados en los últimos años.

La verdadera fortaleza de los PLMs radica en su capacidad para adaptarse a tareas o dominios específicos. Mediante procesos de *fine-tuning*, estos modelos generales pueden especializarse utilizando datos de una tarea concreta. Este proceso resulta fundamental, ya que permite cerrar la brecha entre el conocimiento lingüístico general adquirido durante el preentrenamiento y los requerimientos particulares de una aplicación específica (Anisuzzaman et al., 2025).

El uso de PLM es el punto de partida de este trabajo, ya que permiten aprovechar el conocimiento lingüístico aprendido en idiomas con abundantes recursos, como el español, para facilitar tareas de TA hacia lenguas de bajos recursos, como el quechua.

3.2.1.3 Modelo multilingüe NLLB-200

El modelo *No Language Left Behind* (NLLB-200) fue desarrollado por Meta AI con el propósito de ofrecer traducción automática de alta calidad para más de 200 idiomas, priorizando lenguas con limitados recursos digitales. En su arquitectura principal, el proyecto investigó modelos *Mixture-of-Experts* (MoE) de hasta 54,5 mil millones de parámetros, a partir de los cuales se obtuvieron modelos destilados como *NLLB-200-distilled-600M*, diseñados para conservar la precisión del sistema original pero con un costo computacional viable para su uso práctico (Costa-jussà et al., 2022).

El NLLB-200 es un sistema robusto de TA de propósito general, que ha demostrado una gran capacidad de transferibilidad y adaptabilidad hacia dominios específicos mediante procesos de *fine-tuning* (Costa-jussà et al., 2022).

Gow-Smith y Sánchez Villegas (2023) describen a NLLB-200 como un modelo de última generación específicamente diseñado para entornos de bajos recursos, destacando que fue entrenado en 202 idiomas a lo largo de 1 220 pares de idiomas, incluyendo soporte nativo para el aimara, el guaraní y variantes del quechua.

En este estudio, la variante *NLLB-200-distilled-600M* constituye el modelo base preentrenado, ya que incluye soporte previo para la variante del quechua ayacuchano, lo que facilita enormemente su adaptación específica para la variante del quechua collao.

3.2.1.4 Adaptación de modelos preentrenados

La adaptación de PLM, conocida como *fine-tuning* (ajuste fino), es el proceso mediante el cual un PLM, como un LLM, continúa su entrenamiento en un conjunto de datos específico para un dominio o tarea particular, que es diferente del conjunto de datos original utilizado para entrenar el modelo base, este proceso de adaptación es fundamental para mejorar el rendimiento en tareas específicas y se clasifica dentro de las etapas de post-entrenamiento de los

LLMs (Anisuzzaman et al., 2025).

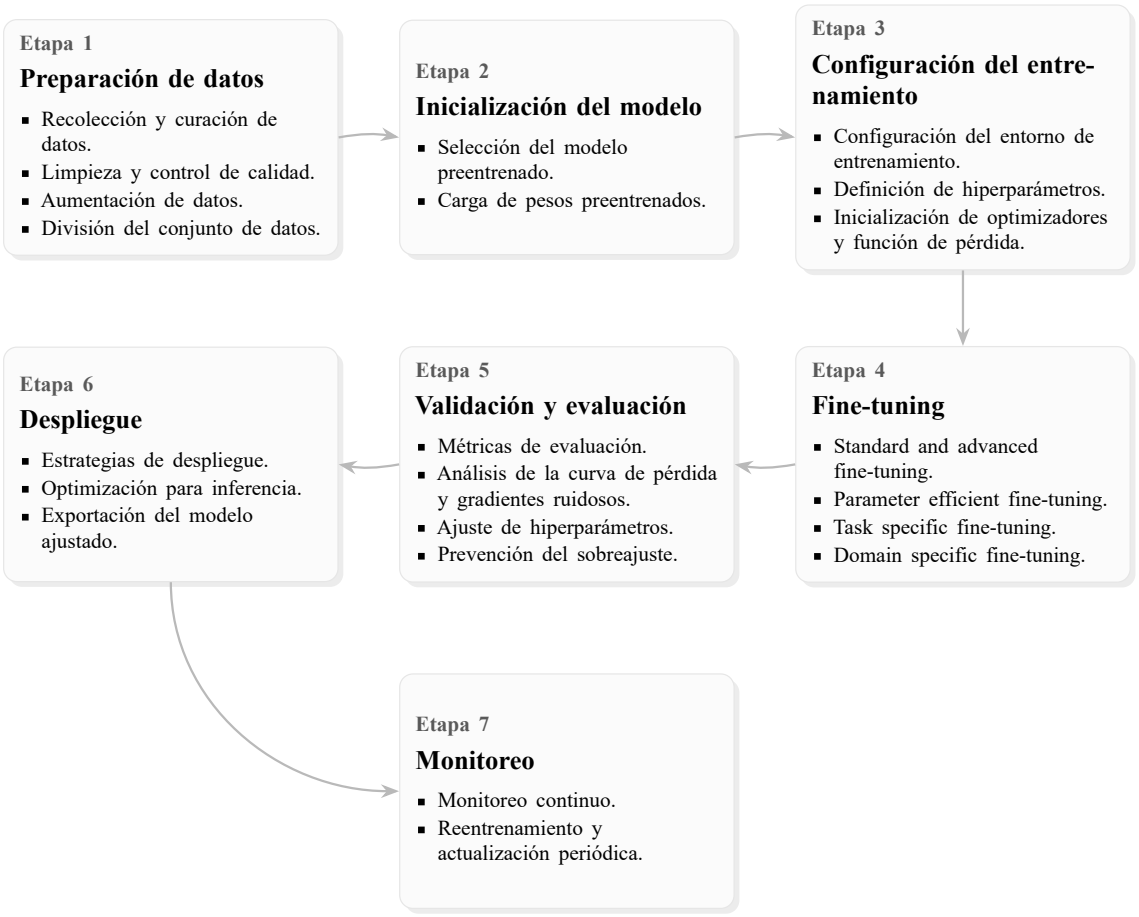
Lu et al. (2025) señalan que la adaptación permite que el modelo conserve las capacidades aprendidas en etapas de entrenamiento anteriores, mientras incorpora conocimiento específico del dominio objetivo.

En la presente investigación, esta etapa es fundamental, ya que permite especializar un modelo generalista en la TA hacia el quechua collao, aprovechando el conocimiento previamente aprendido.

3.2.1.4.1 Etapas de *fine-tuning* para LLMs

El ajuste de un LLM consta de siete etapas clave como se muestra en la Figura 3 y son: preparación de datos, inicialización del modelo, configuración del entrenamiento, *fine-tuning*, validación y evaluación, despliegue y monitoreo. Este proceso adapta el modelo a tareas específicas como la TA.

Figura 3
Etapas del proceso de fine-tuning de los grandes modelos de lenguaje (LLMs)



Nota. Adaptado de Parthasarathy et al. (2024).

3.2.1.4.2 Cantidad de corpus paralelo

La disponibilidad de un corpus paralelo suficientemente amplio y de calidad constituye un requisito fundamental para el éxito de los sistemas de TA, particularmente cuando se trabaja con lenguas de bajos recursos (Chen et al., 2024). Estudios en TAN han demostrado consistentemente que la calidad y cantidad del conjunto de datos de entrenamiento impactan directamente en el desempeño del modelo (Zhong et al., 2024).

Un corpus paralelo más grande permite al modelo capturar correspondencias léxicas, sintácticas y semánticas con mayor precisión, mejorando la fidelidad y coherencia de la traducción. Sin embargo, como señalan Ranathunga et al. (2021), los sistemas de TAN requieren corpus paralelos de gran tamaño, un requisito que no es realista para lenguas de bajos recursos

como el quechua collao.

Por ello, la escasez de corpus paralelos digitales es uno de los principales desafíos para la TA del quechua collao, ya que limita el aprendizaje y desempeño de los modelos, afectando la calidad de las traducciones y haciendo necesario el uso de técnicas que permitan aprovechar mejor los datos disponibles.

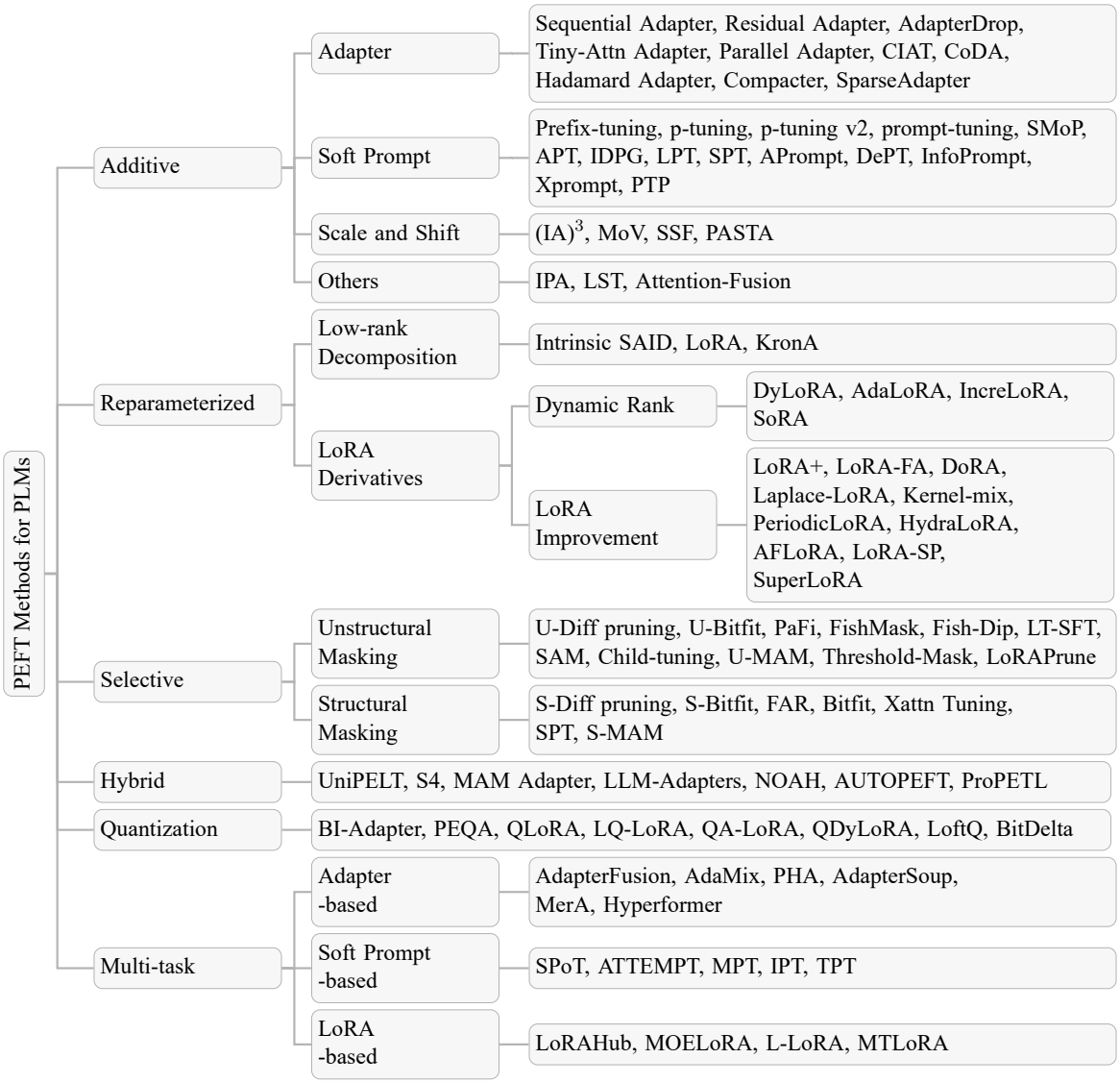
3.2.1.5 Parameter-Efficient Fine-Tuning (PEFT)

Las técnicas de *Parameter-Efficient Fine-Tuning* (PEFT) buscan reducir significativamente la cantidad de parámetros que deben ajustarse durante el entrenamiento, manteniendo congelada la mayor parte de los parámetros preentrenados del LLM. Esto permite adaptar modelos de gran tamaño con menores requerimientos de memoria y capacidad computacional, lo que resulta especialmente útil en entornos con recursos limitados (Parthasarathy et al., 2024).

Las PEFT de última generación han alcanzado niveles de rendimiento comparables al *fine-tuning* completo, algunos métodos comunes de PEFT son *Low-Rank Adaptation* (LoRA), *Prefix Tuning* y *P-Tuning*. La adopción de estos métodos permite un ajuste eficiente de modelos incluso en entornos con recursos limitados (Liu et al., 2025).

La Figura 4 muestra una clasificación general de los métodos de PEFT utilizados en PLMs. La taxonomía organiza las distintas técnicas en varias categorías principales según la forma en que adaptan el modelo, incluyendo métodos aditivos, reparametrizados, selectivos, híbridos, basados en cuantización y enfoques multitarea. Entre los métodos más representativos y ampliamente utilizados destacan LoRA, QLoRA y los enfoques basados en Adapters, debido a su eficiencia y desempeño en tareas de adaptación de LLMs (L. Wang et al., 2025).

Figura 4
Taxonomía de los métodos PEFT



Nota. Adaptado de L. Wang et al. (2025).

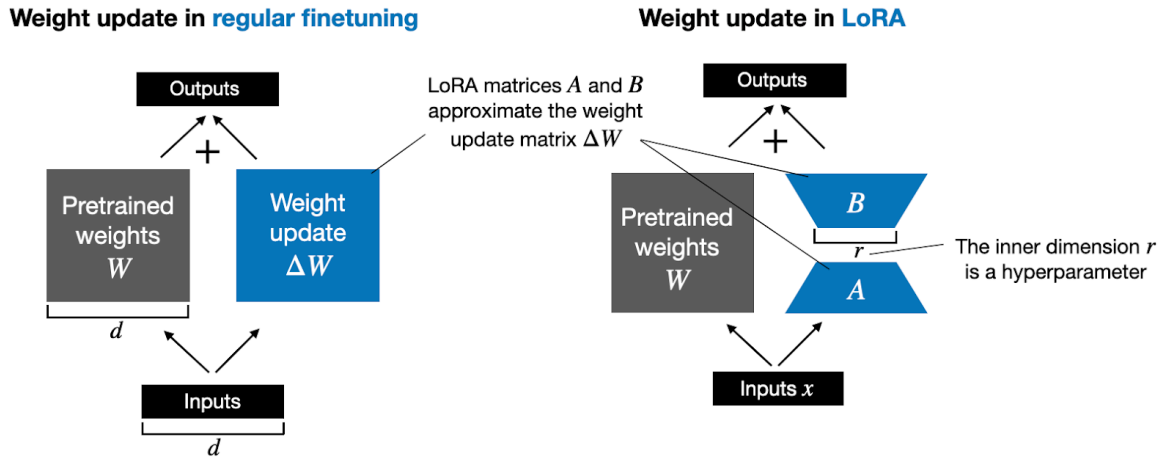
3.2.1.6 Low-Rank Adaptation (LoRA)

LoRA representa un avance significativo en las técnicas PEFT que permite la adaptación eficiente de grandes modelos de lenguaje (LLMs) (Liang et al., 2025). Su objetivo es adaptar LLMs preentrenados a tareas o dominios especializados de manera eficiente (E. Hu et al., 2021). Mediante aproximaciones de bajo rango, LoRA optimiza el consumo de recursos computacionales sin comprometer la capacidad de adaptación del modelo preentrenado (Parthasarathy et al., 2024). La eficiencia de LoRA lo ha posicionado como la técnica PEFT

más utilizada para adaptar LLMs por su eficiencia y resultados a menudo comparables con *fine-tuning* completo (Garcia Gilabert et al., 2024).

La Figura 5 muestra la comparación entre el *fine-tuning* completo y LoRA. En el *fine-tuning* tradicional, el modelo actualiza todos los parámetros preentrenados W mediante una matriz de actualización completa ΔW, lo que implica modificar toda la red (Parthasarathy et al., 2024). En cambio, LoRA congela W y aproxima la actualización ΔW mediante la multiplicación de dos matrices de bajo rango A y B, controladas por un hiperparámetro r, reduciendo significativamente la cantidad de parámetros entrenables (E. Hu et al., 2021). En lugar de almacenar una matriz completa, se aprende una descomposición eficiente tal que $\Delta W \approx AB$.

Figura 5
Una comparación entre las actualizaciones de peso en el fine-tuning regular y el fine-tuning con LoRA



Nota. Reproducido de Raschka (2023).

Por ejemplo, si ΔW tiene $10{,}000 \times 20{,}000$ parámetros (200 millones), y se utiliza un rango $r = 8$, entonces las matrices $A \in \mathbb{R}^{10{,}000\times 8}$ y $B \in \mathbb{R}^{8\times 20{,}000}$ requieren en total $10{,}000 \times 8 + 8 \times 20{,}000 = 240{,}000$ parámetros, lo que representa una reducción de más de 800 veces (E. Hu et al., 2021; Raschka, 2023).

El uso de LoRA es el núcleo de este trabajo, ya que permite realizar el *fine-tuning* del modelo NLLB-200 para la lengua quechua collao entrenando solo una fracción de los

parámetros totales.

3.2.1.6.1 Hiperparámetros de LoRA

En el contexto de LoRA, se consideran dos hiperparámetros fundamentales:

- **Rango (r):** Es un hiperparámetro que controla el equilibrio entre la eficiencia computacional y la capacidad de adaptación, este valor determina la dimensionalidad de las matrices entrenables, un rango muy bajo, incluso uno o dos, es suficiente para la adaptación del modelo (Chen et al., 2024; E. Hu et al., 2021)
- **Factor de escala (α):** Este hiperparámetro regula la magnitud de los ajustes aplicados por las matrices de bajo rango sobre los parámetros del modelo original, es decir, es un parámetro de escalado que ajusta los pesos aprendidos (Garcia Gilabert et al., 2024; E. Hu et al., 2021)

3.2.2 Traducción automática de la lengua quechua

3.2.2.1 Traducción automática (TA)

La TA es un subcampo de la lingüística computacional que busca desarrollar sistemas capaces de traducir texto de un idioma a otro de manera automática (Huarcaya, 2020), empleándose frecuentemente en la literatura internacional el término *Machine Translation* (MT). Sus orígenes se remontan a 1949, cuando Warren Weaver planteó que las computadoras modernas de alta velocidad y capacidad lógica podían emplearse para traducir automáticamente de un idioma a otros (Weaver, 1952). A lo largo de las décadas, la TA ha evolucionado a través de diferentes enfoques.

La primera gran etapa se centró en la TA basada en reglas (Rule-Based Machine Translation o RBMT), la cual surgió en la década de 1950 y 1960, y representa los primeros intentos de traducción automatizada, estos sistemas dependían en gran medida de teorías lingüísticas

y reglas elaboradas a mano por humanos para modelar la gramática y la sintaxis tanto de la lengua de origen como de la lengua de destino (Chen et al., 2024). Este enfoque dependía totalmente del conocimiento humano y demostró ser rígido pero con limitaciones en la precisión.

A finales de los 1980 ocurrió el primer gran salto con la traducción automática estadística (SMT) y se convirtió en el modelo predominante, la SMT se basó en modelos estadísticos que aprendían de grandes corpus de texto paralelo y utilizaban algoritmos de decodificación para traducir (Rivera-Trigueros, 2022). Este paradigma hizo que las máquinas aprendieran el conocimiento de traducción automáticamente a partir de una gran cantidad de datos, en lugar de depender de humanos expertos para escribir reglas (H. Wang et al., 2022). A pesar de su éxito, la SMT requería grandes recursos computacionales y carecía de fundamento lingüístico al estar basada en probabilidades y presentaban problemas con la morfología, especialmente en lenguajes con pocos recursos (Huarcaya, 2020).

Finalmente, en la década de 2010, surgió la TAN, que revolucionó el campo de la TA gracias al uso de técnicas de aprendizaje profundo (*deep learning*). Inicialmente, los modelos de TAN se basaron en arquitecturas de RNN con mecanismos de atención (Wu et al., 2016). Un hito crucial en esta evolución lo marcó Google en 2016 con la introducción de su sistema *Google Neural Machine Translation* (GNMT). Este sistema, basado precisamente en RNN con atención, logró reducir en un 60 % los errores de traducción respecto a los sistemas estadísticos tradicionales, alcanzando una calidad muy cercana a la de traductores humanos en muchos pares de idiomas de altos recursos (Wu et al., 2016). Este éxito impulsó la adopción masiva de la TAN en aplicaciones de gran escala y la consolidó como el paradigma dominante, superando a las técnicas convencionales de RBMT y SMT (Huarcaya, 2020). Sin embargo, el avance más significativo llegó con la propuesta de la arquitectura Transformer por Vaswani et al. (2023).

3.2.2.1.1 Traducción automática neuronal (TAN)

La TAN ha evolucionado a través de varias arquitecturas que marcaron hitos en su desarrollo (Stahlberg, 2020) y actualmente es el paradigma dominante (Rivera-Trigueros, 2022).

Los primeros sistemas estuvieron basados en modelos secuencia a secuencia (seq2seq) construidos con RNN, que utilizaban una arquitectura codificador-decodificador, que permitían codificar una oración y generar su traducción de manera secuencial (Nguyen, 2019). Estos podían producir traducciones razonables para frases cortas, pero se desmoronaban al aumentar la longitud de las oraciones. (Koehn, 2017)

Posteriormente, la incorporación del mecanismo de atención permitió superar las limitaciones de las RNN en oraciones largas, al reemplazar el vector de contexto fijo por uno variable, mejorando la alineación entre palabras fuente y destino (Muñoz, 2021).

Finalmente, la aparición del modelo Transformer en 2017 marcó un cambio de paradigma, ya que está basado completamente en mecanismos de autoatención (*self-attention*), permitiendo que el codificador y el decodificador consideren cada palabra de la secuencia de entrada en paralelo y prescinde totalmente de la recurrencia y las convoluciones (Vaswani et al., 2023). Esta dependencia total en la atención (específicamente, la autoatención y la atención de múltiples cabezas, o *multi-head attention*) permite una mayor paralelización del entrenamiento y, en general, conduce a una calidad superior de traducción (Stahlberg, 2020). El Transformer representó un salto cualitativo, logrando el estado del arte y se consolidó como la arquitectura dominante en la TA actual, superando ampliamente a las RNN y otras variantes como las CNN en calidad de traducción (J. Hu, 2024).

La TAN constituye el núcleo tecnológico de este trabajo, ya que permite que sistemas como NLLB-200 capturen patrones lingüísticos complejos del quechua collao sin depender de reglas gramaticales diseñadas manualmente.

3.2.2.2 La lengua quechua

El quechua es reconocido como una familia lingüística con presencia en siete países sudamericanos: Perú, Ecuador, Colombia, Bolivia, Argentina, Chile y Brasil, es considerada la lengua originaria con mayor predominancia en Sudamérica (Torero, 2002). En Perú, de acuerdo con el Censo Nacional de 2017 se revelaron que, el 22,3 % de la población de 12 y más años de edad, que es equivalente a 5 millones 176 mil 809 personas se autoidentificó como Quechua (INEI, 2017).

El quechua es clasificado como una lengua vital, sin embargo, a pesar de su vitalidad general, los estudios lingüísticos señalan que el quechua constituye una familia lingüística diversa con múltiples lenguas y no una sola (Ministerio de Educación, 2018). La familia lingüística quechua se organiza tradicionalmente en dos grandes grupos, tal como se observa en la Tabla 2, denominados Quechua I (QI), y Quechua II (QII) (Cerron-Palomino, 2003). No obstante, se reconocen cuatro ramas principales en Perú: Quechua Sureño, Quechua Central, Quechua Norteño y Quechua Amazónico, en la rama de quechua sureño se divide en dos variantes principales: Quechua Chanka (Ayacuchano) y Quechua Collao (Cuzqueño) (Ministerio de Educación, 2018).

Lingüísticamente, el quechua es una lengua aglutinante y polisintética, cuyas palabras suelen estar formadas por múltiples morfemas que expresan información gramatical y semántica. Esta complejidad morfológica genera una gran variedad de formas léxicas y constituye uno de los principales desafíos para las tareas de procesamiento del lenguaje natural y traducción automática (Chen et al., 2024).

3.2.2.3 Calidad de traducción

La calidad de la traducción es un concepto relativo y complejo que hace referencia tanto al producto traducido como al proceso de traducción. En el ámbito de la TA, su evalua-

Tabla 2
Variedades del quechua en el Perú

Rama	Variedad	Departamento donde se habla	Estado de vitalidad
Quechua amazónico	Kichwa amazónico: Pastaza, Napo, Putumayo, Tigre, Alto Napo (Santarrosino–Madre de Dios), Chachapoyas y San Martín	Loreto, Madre de Dios, Chachapoyas y San Martín	En peligro
Quechua norteño	Quechua Cajamarca	Cajamarca	Seriamente en peligro
	Quechua Inkawasi Kañaris	Lambayeque y Piura (Comunidad de Chilcapampa, distrito de Huarmaca, provincia de Huancabamba; centro poblado La Pilca, distrito de Buenos Aires, provincia de Morropón)	Vital
Quechua central	Quechua Pataz	La Libertad	Vital
	Quechua Cajatambo, Oyón, Huaura	Lima	Seriamente en peligro
	Quechua Yauyas		Seriamente en peligro
	Quechua Áncash	Áncash	Vital
	Quechua Huánuco	Huánuco	En peligro
	Quechua Pasco	Pasco	Seriamente en peligro
	Quechua Wanka	Junín	Seriamente en peligro
Quechua sureño	Quechua Chanka	Huancavelica, Ayacucho y Apurímac (Andahuaylas, Aymaraes y Chincheros)	Vital
	Quechua Collao	Apurímac (Abancay, Grau, Antabamba y Cotabambas), Cusco, Puno, Arequipa y Moquegua	Vital

Nota. La clasificación se organiza en cuatro ramas del quechua (amazónico, norteño, central y sureño), indicando sus variedades, los departamentos donde se hablan y su estado de vitalidad lingüística. Fuente: Ministerio de Educación (2018).

ción se realiza mediante un enfoque pragmático que combina la evaluación humana con el uso de métricas automáticas, permitiendo valorar el desempeño de los sistemas de traducción (Do Campo Bayón, 2023).

Asimismo, Rivera-Trigueros (2022) señalan que la calidad de una TA puede determinarse a partir del grado de similitud entre la traducción generada por el sistema y una traducción de referencia elaborada por un traductor humano, considerando que una mayor coincidencia refleja un mejor desempeño del modelo.

En la presente investigación, la calidad de la traducción permite evaluar el efecto de la adaptación del modelo multilingüe NLLB-200 mediante la técnica LoRA en la traducción automática del español al quechua collao. Para ello, se utilizan métricas automáticas que permiten evaluar la calidad de las traducciones generadas por el modelo mediante su comparación con traducciones de referencia.

3.2.2.4 Métricas de evaluación de sistemas de traducción automática

La evaluación de sistemas de TA es un proceso esencial para proporcionar una estimación cuantitativa de la calidad de las traducciones generadas (Haddow et al., 2022). Para ello, se emplean métricas de evaluación que permiten a investigadores y desarrolladores medir objetivamente el rendimiento de estos modelos (Chauhan & Daniel, 2023).

Entre los distintos enfoques de evaluación, la evaluación automática es la preferida en la práctica, ya que es considerablemente más rápida, fácil de implementar y económica que la evaluación humana, la cual requiere de evaluadores bilingües expertos, lo que resulta en baja eficiencia y un alto costo (Mondal et al., 2023).

Entre la gran variedad de métricas automáticas existentes, algunas de las más utilizadas para la evaluación de TA son las métricas basadas en léxicos, se basan en la comparación de los elementos léxicos (por ejemplo, palabras o frases) de la hipótesis con los de una traducción de referencia (Lee et al., 2023). Las más utilizadas son las siguientes:

3.2.2.4.1 BLEU

La métrica *Bilingual Evaluation Understudy* (BLEU) es un método automático, rápido y económico de evaluación de la TA, propuesto como un ”sustituto automatizado” de los jueces humanos capacitados. La calidad de una TA se mide por qué tan cerca se encuentra de una o más traducciones humanas profesionales de referencia. Para cuantificar esta cercanía, BLEU compara la traducción candidata del sistema con las referencias contando la cantidad de coincidencias de palabras y frases cortas (n-gramas) (Papineni et al., 2002).

Por su parte, Lee et al. (2023) señalan que el BLEU se calcula a partir de tres componentes principales: (1) la precisión basada en n-gramas entre la traducción generada y la traducción de referencia, (2) una penalización por brevedad (*Brevity Penalty*, BP) para evitar favorecer traducciones excesivamente cortas y (3) el recorte (*clipping*), que limita el conteo de n-gramas repetidos. La precisión se obtiene al dividir el número de n-gramas coincidentes entre el número total de n-gramas generados.

A pesar de su popularidad, BLEU presenta algunas limitaciones al evaluar traducciones al quechua, ya que su enfoque basado en palabras completas puede penalizar variaciones morfológicas correctas que son frecuentes en las lenguas aglutinantes (Mager et al., 2023).

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

- BLEU: Métrica BLEU final que mide la calidad de la traducción automática.
- BP: *Brevity Penalty*, penalización por brevedad aplicada cuando la hipótesis es más corta que la referencia.
- N: Máximo nivel de n-gramas considerados (típicamente $N = 4$ en BLEU-4).
- w_n: Peso asignado a cada precisión de n-gramas; en BLEU-4 se usa $w_n = \frac{1}{N}$.
- p_n: Precisión modificada de n-gramas (proporción de n-gramas de la hipótesis que

aparecen en la referencia, aplicando recorte por frecuencia).

3.2.2.4.2 TER

La métrica *Translation Edit Rate* (TER) es una medida intuitiva y ampliamente utilizada para la evaluación automática de la calidad de la salida de la TA. TER calcula la cantidad mínima de operaciones de edición (desplazamiento, inserción, eliminación y sustitución) que un humano tendría que ejecutar para lograr la coincidencia exacta con la referencia (Snover et al., 2006).

Asimismo, Lee et al. (2023) indican que la TER evalúa la calidad de una traducción automática calculando la cantidad mínima de ediciones necesarias para hacerla coincidir con una traducción de referencia; cuanto menor es su valor, mejor es la traducción.

Sin embargo, la TER presenta varias limitaciones que limitan su capacidad para evaluar con precisión el rendimiento de la traducción automática, ya que se basa principalmente en la coincidencia de palabras entre la traducción generada y la referencia, sin considerar completamente la similitud semántica ni la fluidez del texto (Lee et al., 2023).

$$\text{TER} = \frac{E}{R}$$

- E = Número mínimo de ediciones necesarias para transformar la hipótesis en la referencia.
    - Inserciones
    - Eliminaciones
    - Sustituciones
    - Desplazamientos (shifts)
- R = Longitud de la referencia, es decir, el número total de palabras en la oración de referencia.

3.2.2.4.3 METEOR

La métrica *Metric for Evaluation of Translation with Explicit ORdering* (METEOR) es una métrica automática para la evaluación de la TA que fue desarrollada para ofrecer una mejor correlación con los juicios humanos de calidad en comparación con métricas anteriores como BLEU, se fundamenta en un concepto generalizado de coincidencia de unigramas (*unigram matching*) entre la traducción producida por el sistema y las traducciones de referencia humanas (Banerjee & Lavie, 2005).

$$\text{METEOR} = F_\alpha \cdot (1 - \text{Penalidad})$$

$$F_\alpha = \frac{P \cdot R}{\alpha \cdot P + (1 - \alpha) \cdot R}$$

- P: Precisión entre los segmentos alineados (porcentaje de palabras alineadas respecto a la hipótesis).
- R: Recall entre los segmentos alineados (porcentaje de palabras alineadas respecto a la referencia).
- F_α: Media armónica ponderada entre precisión y recall.
- α: Parámetro que controla el peso entre precisión y recall (típicamente 0.9).
- m: Número total de palabras alineadas.
- ch: Número de fragmentos (chunks) en la alineación; un chunk es una secuencia de palabras alineadas consecutivamente y en el mismo orden.
- γ, β: Parámetros de ajuste usados para calcular la penalización.
- Penalidad: Factor que reduce la puntuación cuando el orden de las palabras difiere entre hipótesis y referencia.

3.2.2.4.4 ChrF++

El *Character F-Score Plus* (ChrF++) es una métrica de evaluación automática de la TA diseñada para combinar los beneficios de las métricas basadas en caracteres y las basadas en palabras, el puntaje ChrF++ se calcula obteniendo el F-score tanto para los n-gramas de caracteres como para los n-gramas de palabras, y luego promediando estos resultados, ChrF++ es una versión mejorada de la métrica original *Character F-Score* (ChrF) (Popović, 2017).

Asimismo, Mager et al. (2023) señalan que, en lenguas morfológicamente ricas como el quechua, las métricas basadas en caracteres, como ChrF, suelen ser más adecuadas que las métricas basadas únicamente en palabras, debido a que ofrecen una mayor flexibilidad para capturar las variaciones morfológicas propias de estas lenguas. En ese sentido, la presente investigación también emplea ChrF++, por tratarse de una extensión de ChrF que incorpora información basada en n-gramas de palabras, proporcionando una evaluación más completa del desempeño del modelo.

Por ello, ChrF++ se considera una de las métricas más adecuadas para esta investigación, ya que permite evaluar con mayor precisión las traducciones al quechua collao, una lengua caracterizada por su compleja estructura morfológica.

$$\text{ChrF}^{++} = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R}$$

donde la precisión y exhaustividad combinan información de caracteres y palabras:

$$P = \alpha \cdot P_c + (1 - \alpha) \cdot P_w$$

$$R = \alpha \cdot R_c + (1 - \alpha) \cdot R_w$$

- P: Precisión combinada (caracteres + palabras).

- R: Exhaustividad combinada (caracteres + palabras).
- P_c: Precisión basada en n-gramas de caracteres.
- R_c: Exhaustividad basada en n-gramas de caracteres.
- P_w: Precisión basada en n-gramas de palabras.
- R_w: Exhaustividad basada en n-gramas de palabras.
- β: Factor que pondera la importancia de la exhaustividad sobre la precisión (por defecto, $\beta = 2$).
- α: Peso que asigna la contribución de los caracteres frente a las palabras (por defecto, $\alpha = 0.5$).

3.2.2.5 Eficiencia computacional en sistemas de traducción automática

La eficiencia computacional en el contexto de la TA, se define como la capacidad de los modelos para lograr bajo costo y alta eficiencia en sus operaciones (H. Wang et al., 2022).

La eficiencia computacional es un factor determinante en el diseño y uso de sistemas de TAN, particularmente cuando se emplean modelos basados en arquitecturas Transformer y modelos multilingües de gran escala, si bien los Transformers permiten un alto grado de paralelización y mejoran sustancialmente la calidad de la traducción, su complejidad computacional y consumo de recursos aumentan significativamente conforme crece el tamaño del modelo y la longitud de las secuencias procesadas (Vaswani et al., 2023).

En el contexto de lenguas de bajos recursos, como el quechua collao, esta problemática se acentúa, ya que el entrenamiento y la adaptación de modelos de gran escala requiere una gran infraestructura, que es inaccesible en entornos con recursos limitados, restringiendo la capacidad de investigadores y desarrolladores locales para construir soluciones lingüísticas, en este sentido, la literatura reciente resalta la importancia de evaluar no solo la calidad de traducción alcanzada, sino también su eficiencia computacional y los costos asociados a su entrenamiento y adaptación (Zhong et al., 2024).

En esta investigación, la eficiencia computacional es un aspecto fundamental, pues permite adaptar modelos de gran escala, como NLLB-200 en sus versiones de 600M, 1.3B y 3.3B parámetros, utilizando hardware accesible. Esto hace posible desarrollar y evaluar sistemas de TA para el quechua collao en entornos con recursos computacionales limitados.

3.2.2.5.1 Tiempo total de entrenamiento

El tiempo total de entrenamiento en los sistemas de TAN está determinado por factores como el número de épocas, el tamaño del lote (*batch size*), y la complejidad del modelo (Do Campo Bayón, 2023). En el caso del *fine-tuning* de modelos preentrenados demanda un menor número de tiempo y memoria que el entrenamiento desde cero de un LLM, ya que se reutiliza las representaciones linguisticas previamente aprendidas (Anisuzzaman et al., 2025).

No obstante, en arquitecturas basadas en Transformer, el costo computacional sigue siendo elevado debido a la complejidad del mecanismo de atención, que crece cuadráticamente con la longitud de las secuencias (Vaswani et al., 2023).

Para esta investigación, reducir el tiempo de entrenamiento es fundamental para hacer viable la adaptación de modelos de gran escala en entornos con recursos limitados. En este sentido, técnicas como LoRA permiten acelerar el *fine-tuning* al entrenar únicamente una pequeña parte de los parámetros del modelo original.

3.2.2.5.2 Uso de memoria GPU

El uso de memoria GPU representa una de las principales limitaciones en la adaptación de modelos de gran escala, dado que los requerimientos de memoria suelen exceder la capacidad de los equipos que posee la mayoría de las personas (Parthasarathy et al., 2024). Además, el *fine-tuning* completo implica actualizar la totalidad de los parámetros del modelo, lo que incrementa significativamente el consumo de memoria (Chen et al., 2024).

Para mitigar este problema, las técnicas de PEFT permiten entrenar solo una fracción

del modelo, manteniendo congelados los pesos originales (Anisuzzaman et al., 2025). En particular, LoRA introduce adaptaciones de bajo rango que reducen el número de parámetros entrenables y, en consecuencia, el uso de memoria GPU, sin afectar el rendimiento del modelo (E. Hu et al., 2021).

En esta investigación, el uso de memoria GPU es un factor importante, ya que determina la viabilidad de adaptar modelos NLLB-200 en entornos con recursos limitados. Por ello, se emplean técnicas como LoRA, que reducen los requerimientos de memoria durante el *fine-tuning*.

3.3 Definición de términos

AmericasNLP

Se refiere a un taller (*workshop*) y una tarea compartida (*Shared Task*) enfocada en el PLN y la TA para las lenguas nativas americanas (Ahmed et al., 2023).

Corpus paralelo

En una colección de textos alineados en dos idiomas y es esencial para entrenar y evaluar un sistema de TAN (Do Campo Bayón, 2023).

Dataset

Un dataset es un conjunto de datos que se encuentra organizado de manera estructurada, generalmente en tablas, matrices o formatos como CSV y JSON. Esta organización permite almacenar, consultar y analizar la información de una forma más ordenada. Los datasets constituyen un recurso fundamental en tareas de análisis de datos, aprendizaje automático (*machine learning*, ML), IA y otras aplicaciones en las que se requiere disponer de información accesible y adecuada para su procesamiento (Badman & Kosinski, s.f.).

Datos de entrenamiento

Corresponden al subconjunto de datos utilizado durante el proceso de entrenamiento de un modelo (Google for Developers, s.f.).

Datos de validación

Es el subconjunto de datos utilizado para realizar una evaluación inicial del modelo entrenado. Generalmente, se utiliza durante el desarrollo para evaluar y ajustar el modelo antes de realizar la evaluación con el conjunto de prueba (Google for Developers, s.f.).

Datos de prueba

Es un subconjunto del conjunto de datos reservado para probar un modelo entrenado (Google for Developers, s.f.).

Inteligencia artificial (AI)

Constituye un conjunto de tecnologías que permite a las computadoras aprendan, piensen y realicen tareas que antes solo podían realizar las personas, como entender el lenguaje, analizar información y dar recomendaciones (Google Cloud, s.f.).

Google Colab

Es un servicio basado en Jupyter Notebook que puede utilizarse sin una configuración previa y ofrece acceso gratuito a recursos de cómputo, como GPU y TPU. Su uso es común en actividades relacionadas con el aprendizaje automático, la ciencia de datos y la educación (Google Colab, s.f.).

GPU

Las unidades de procesamiento gráfico (GPU) son una pieza clave en la tecnología actual. Gracias a su capacidad para realizar muchos cálculos y procesamientos al mismo tiempo, permiten que los programas funcionen más rápido y que podamos desarrollar aplicaciones más eficientes, potentes y con mejores resultados visuales. (Lenovo, s.f.)

Grandes modelos de lenguaje (LLMs)

Son modelos entrenados con grandes cantidades de datos y pueden tener cientos de miles de millones de parámetros, lo que los hace capaces de comprender y generar lenguaje humano (AWS Amazon, s.f.).

Hugging Face

Es una plataforma donde la comunidad de aprendizaje automático colabora en modelos, conjuntos de datos y aplicaciones (Hugging Face, s.f.).

Modelo multilingue

En el contexto de la TAN, son sistemas diseñados para gestionar la traducción entre múltiples idiomas utilizando un único modelo (Haddow et al., 2022).

Lengua de bajos recursos

Es aquella que padece una escasez significativa de recursos lingüísticos, tales como corpora, diccionarios y conjuntos de datos anotados, estas lenguas suelen carecer de un apoyo digital suficiente, lo que obstaculiza su investigación y preservación (Zhong et al., 2024).

Pipeline

Un pipeline de *machine learning* es un flujo de trabajo estructurado que organiza de manera secuencial las etapas necesarias para diseñar, desarrollar y desplegar un modelo de aprendizaje automático. Su propósito es sistematizar el proceso de construcción del modelo, facilitando una implementación más eficiente, reproducible y organizada (Belcic & Stryker, s.f.).

Procesamiento del lenguaje natural (PLN)

Es una rama de la IA que, mediante técnicas de aprendizaje automático, enseña a las máquinas a interpretar, generar y responder en lenguaje humano de forma natural (IBM, s.f.).

RunPod

Es una plataforma de computación en la nube especializada en IA que proporciona acceso bajo demanda a GPUs de alto rendimiento, permitiendo a los desarrolladores ejecutar desde experimentos hasta aplicaciones en producción sin necesidad de gestionar infraestructura propia (Runpod, s.f.).

IV. Metodología

4.1 Tipo y nivel de investigación

La presente investigación se desarrolló bajo el enfoque cuantitativo, el cual se caracteriza por seguir un proceso secuencial para probar hipótesis (Hernández-Sampieri & Mendoza Torres, 2023).

En concordancia con dicho enfoque, la investigación es de tipo aplicada, dado que, según Hernández-Sampieri y Mendoza Torres (2023), este tipo de investigación busca utilizar el conocimiento científico para resolver problemas concretos en contextos reales. En este sentido, el trabajo emplea fundamentos del PLN para mejorar la TA de la lengua quechua collao mediante la adaptación del modelo NLLB-200 con la técnica LoRA.

La investigación es de nivel explicativo, ya que su objetivo es establecer las causas del fenómeno estudiado. De acuerdo con Hernández-Sampieri y Mendoza Torres (2023), este alcance busca explicar por qué ocurre un evento y en qué condiciones se manifiesta. En este estudio, se analiza el efecto de la adaptación del modelo NLLB-200 mediante la técnica LoRA sobre su desempeño en la traducción automática del español al quechua collao.

En cuanto al diseño metodológico, se empleó un diseño preexperimental con preprueba y posprueba de un solo grupo. Este diseño permite evaluar los cambios producidos después de aplicar una intervención sobre una misma unidad de análisis. Según Hernández-Sampieri y Mendoza Torres (2023), los diseños preexperimentales con preprueba y posprueba de un solo grupo consisten en realizar una medición antes de la intervención y otra después de su aplicación, con el propósito de identificar los cambios generados.

En la presente investigación, la preprueba corresponde a la evaluación del desempeño del modelo base NLLB-200 mediante las métricas de traducción automática establecidas. Posteriormente, se aplicó como intervención el proceso de adaptación del modelo utilizando la técnica LoRA con un corpus paralelo español–quechua collao. Finalmente, la posprueba consistió en evaluar el desempeño del modelo adaptado utilizando las mismas métricas y el mismo conjunto de prueba, lo que permitió analizar el efecto de la adaptación realizada.

4.2 Ámbito temporal y espacial

4.2.1 Espacial

El estudio se enfocó en la lengua quechua collao, la variante hablada principalmente en la región de Cusco, en el sur andino del Perú, por lo tanto, se excluyeron otras variantes del quechua presentes en el país y en el extranjero.

4.2.2 Temporal

El desarrollo de la presente investigación tuvo lugar durante el año 2026, tiempo en el que se efectuó la recolección, tratamiento y análisis del corpus paralelo español–quechua collao, así como la implementación y evaluación del *fine-tuning* del modelo NLLB-200 mediante la técnica LoRA.

4.3 Población y muestra

4.3.1 Población

La población de estudio estuvo conformada por los corpus paralelos español–quechua collao disponibles en fuentes abiertas y repositorios digitales utilizados en investigaciones de PLN y TAN. Entre las principales fuentes utilizadas se encuentran el conjunto de datos JW300 (Agić & Vulić, 2019), el cual reúne textos multilingües de los testigos de Jehová,

en nuestro caso, empleamos las versiones en español y en quechua collao. También se tiene de investigaciones que generaron corpus paralelos de español-quechua collao, entre los que destaca el de Paccotacya Yanque y Huanca Anquise (2022). Estas fuentes fue complementada con datasets publicados en Hugging Face. Algunos corpus estaban estructurados como corpus paralelos, conformados por pares de oraciones alineadas entre ambos idiomas, mientras que otros fueron obtenidos en archivos separados por idioma en formato TXT, en estos últimos casos, fue necesario verificar la correspondencia línea por línea entre las oraciones en español y sus respectivas traducciones en quechua collao. La combinación de estas fuentes permitió recopilar un total de 139 278 pares de oraciones paralelas, tal como se presenta en la Tabla 3.

Tabla 3
Fuentes del corpus paralelo español-quechua collao

Fuente	Pares de oraciones
JW300 (Agić & Vulić, 2019)	136 589
Paccotacya Yanque & Huanca Anquise (2022)	2 070
Hugging Face	619
Total	**139 278**

4.3.2 Muestra

De acuerdo con Hernández-Sampieri y Mendoza Torres (2023), en la ruta cuantitativa una muestra constituye un subgrupo de la población del cual se recolectan los datos necesarios para responder al problema de investigación. Asimismo, Hernández-Sampieri y Mendoza Torres (2023) señalan que, aunque generalmente se trabaja con muestras por razones de tiempo y recursos, en algunos casos es posible estudiar todos los elementos de la población mediante un censo.

En esta investigación, la unidad de análisis estuvo conformada por pares de oraciones paralelas español–quechua collao, debido a que el estudio se desarrolla en el ámbito del PLN. No se realizó un proceso de muestreo, ya que se trabajó con el corpus paralelo dis-

ponible recopilado de las fuentes seleccionadas. Luego de aplicar los procesos de limpieza, normalización y filtrado, se obtuvo un corpus final de 126,964 pares de oraciones, el cual fue utilizado para el entrenamiento, validación y evaluación del modelo NLLB-200, según se presenta en la Tabla 4.

Tabla 4
Distribución del corpus

Conjunto	Cantidad de pares	Porcentaje
Entrenamiento	101 570	80 %
Validación	12 697	10 %
Prueba	12 697	10 %
Total	**126 964**	**100 %**

4.4 Instrumentos

Los recursos utilizados en la investigación se clasificaron en herramientas computacionales e instrumentos de medición. Las herramientas computacionales permitieron realizar el procesamiento del corpus y el *fine-tuning* del modelo neuronal, mientras que los instrumentos de medición fueron empleados para evaluar objetivamente el desempeño del modelo mediante métricas de TA.

En las primeras etapas del procesamiento se utilizó Google Colab, debido a su facilidad para trabajar con notebooks interactivos y bibliotecas relacionadas con PLN. Posteriormente, debido a los requerimientos computacionales del modelo NLLB-200, las etapas de entrenamiento y evaluación fueron trasladadas hacia RunPod utilizando instancias GPU de mayor capacidad.

Para el desarrollo del experimento se utilizó el lenguaje de programación Python junto con bibliotecas especializadas como Transformers, PEFT, Datasets y Pandas. La evaluación del modelo se realizó mediante las métricas BLEU, ChrF y ChrF++, calculadas utilizando las bibliotecas Evaluate y SacreBLEU. En la Tabla 5 se presentan los principales recursos

empleados durante la investigación.

Tabla 5
Herramientas computacionales e instrumentos de medición utilizados en la investigación

Tipo	Recurso	Función principal
Herramienta computacional	Google Colab	Preprocesamiento del corpus y ejecución de notebooks.
Herramienta computacional	RunPod	Entrenamiento y evaluación del modelo utilizando GPU de alto rendimiento.
Herramienta computacional	Python	Desarrollo de scripts para el procesamiento del corpus y entrenamiento del modelo.
Herramienta computacional	Hugging Face Transformers	Carga del modelo NLLB-200, tokenización y ejecución del *fine-tuning*.
Herramienta computacional	PEFT	Implementación de la técnica LoRA para el *fine-tuning*.
Herramienta computacional	Datasets	Gestión y carga de los conjuntos de entrenamiento, validación y prueba.
Herramienta computacional	Pandas	Manipulación y organización de los datos.
Instrumento de medición	BLEU	Evaluación automática de la calidad de las traducciones mediante coincidencia de *n*-gramas.
Instrumento de medición	ChrF++	Evaluación automática basada en coincidencias de caracteres y palabras.

4.5 Procedimientos

El procesamiento de los datos y la ejecución del experimento se realizaron mediante un procedimiento secuencial compuesto por diversas etapas, las cuales abarcan desde la obtención y preparación del corpus paralelo hasta la evaluación del modelo adaptado. Este procedimiento comprende la obtención e integración de los corpus, la limpieza y normalización textual, el filtrado por longitud, la división del conjunto de datos en entrenamiento, validación y prueba, la tokenización, la configuración de la técnica LoRA, el proceso de *fine-tuning* del modelo NLLB-200 y, finalmente, la evaluación automática mediante las métricas BLEU, ChrF y ChrF++. El flujo completo del procedimiento experimental se presenta en la

Figura 6.

Figura 6
Flujo del procedimiento experimental



Nota. Elaboración propia del flujo del procedimiento experimental para la adaptación del modelo NLLB-200 mediante LoRA.

4.5.1 Integración y consolidación de corpus

Los corpus utilizados durante la investigación provinieron de diferentes fuentes y formatos. Algunos datasets fueron descargados desde Hugging Face, mientras que otros se encontraban organizados manualmente en archivos separados por idioma. Debido a ello, fue necesario realizar un proceso previo de integración y consolidación de datos antes de iniciar el entrenamiento del modelo.

En el caso de los archivos paralelos en formato TXT, un archivo contenía las oraciones

en español y otro sus respectivas traducciones en quechua collao. Para evitar errores durante el entrenamiento, primero se verificó que ambas listas mantuvieran la misma cantidad de líneas y que las traducciones conservaran correspondencia entre sí.

Posteriormente, los corpus fueron convertidos a estructuras tabulares utilizando la biblioteca *pandas*, con el propósito de facilitar la manipulación, limpieza, filtrado y organización de los pares paralelos durante las etapas de preprocesamiento y entrenamiento del modelo. La Tabla 6 muestra un ejemplo simplificado de la estructura tabular generada.

Tabla 6
Ejemplo de estructura tabular del corpus paralelo

spa	**quz**
Mira la casa	Wasita qhaway
No hay nadie	Mana pipas kanchu
Todos los varones están trabajando	Llapan qharikuna llank´ashanku

Nota. Ejemplo ilustrativo del corpus paralelo organizado mediante *pandas*.

Figura 7
Estructuración tabular del corpus

```
import pandas as pd

df = pd.DataFrame({
    "spa": spanish_lines,
    "quz": quechua_lines
})
```

Asimismo, algunos datasets descargados desde Hugging Face ya se encontraban organizados en formato paralelo. Sin embargo, fue necesario convertirlos al mismo formato utilizado por el resto de corpus para unificar el pipeline de procesamiento. Después de integrar todas las fuentes, se obtuvo un único corpus consolidado que posteriormente fue utilizado durante las etapas de limpieza, tokenización y entrenamiento.

Tabla 7
Fuentes integradas en el corpus consolidado

Fuente	Tipo de archivo	Observaciones
JW300 (Agić & Vulić, 2019)	TXT paralelo	Alineación manual
Paccotacya Yanque & Huanca Anquise (2022)	CSV	Limpieza parcial
Corpus adicional	Hugging Face	Filtrado previo

Nota. Resumen de las principales fuentes utilizadas para la construcción del corpus paralelo consolidado.

4.5.2 Limpieza y normalización textual

Después de consolidar el corpus paralelo, se realizó una etapa de limpieza textual con el propósito de reducir inconsistencias presentes en los datos. Esta fase fue importante debido a que algunos registros contenían caracteres extraños, espacios duplicados o líneas vacías que podían afectar el entrenamiento del modelo.

En primer lugar, se aplicó normalización Unicode NFC para reducir inconsistencias relacionadas con caracteres acentuados y diferencias de codificación presentes en algunos textos, con lo que se evita inconsistencias durante la tokenización y el entrenamiento del modelo.

Tabla 8
Ejemplos de normalización Unicode NFC

Texto original	Resultado normalizado
nin\u0303o (tilde combinante)	niño
cancio\u0301n (acento agudo combinante)	canción
pingu\u0308ino (diéresis combinante)	pingüino

Nota. La normalización Unicode NFC estandariza la codificación de caracteres acentuados y otros signos diacríticos para garantizar una representación uniforme del texto.

Figura 8
Algoritmo para la normalización Unicode NFC

```
import unicodedata

df["spa"] = df["spa"].astype(str)
df["quz"] = df["quz"].astype(str)

df["spa"] = df["spa"].apply(
    lambda x: unicodedata.normalize("NFC", x)
)

df["quz"] = df["quz"].apply(
    lambda x: unicodedata.normalize("NFC", x)
)
```

Posteriormente, se aplicó normalización de espacios utilizando expresiones regulares para corregir problemas relacionados con tabulaciones, múltiples espacios y saltos de línea innecesarios.

Tabla 9
Ejemplos de normalización de espacios y caracteres invisibles

Texto original	Resultado procesado
el\tcongreso\n\napeló	el congreso apeló
hola\xa0mundo	hola mundo
texto\u200binvisible	texto invisible
muchos espacios	muchos espacios

Nota. Ejemplo de eliminación de caracteres invisibles y normalización de espacios múltiples.

Figura 9
Algoritmo para la normalización de espacios y caracteres invisibles en el corpus

```
ESPACIOS_ESPECIALES =
↪  re.compile(r"[\t\n\r\xa0\u200b\u200c\u200d\u2009\u202f\u3000]")
ESPACIOS_MULTIPLES = re.compile(r" {2,}")

def normalizar_espacios(texto: str) -> str:
    texto = ESPACIOS_ESPECIALES.sub(" ", texto)
    texto = ESPACIOS_MULTIPLES.sub(" ", texto)
    return texto.strip()

df["spa"]  = df["spa"].progress_apply(normalizar_espacios)
df["quz"] = df["quz"].progress_apply(normalizar_espacios)
```

Nota. El procedimiento elimina caracteres invisibles y normaliza espacios múltiples para evitar inconsistencias durante el preprocesamiento y la tokenización.

A continuación, se eliminaron los registros vacíos y las líneas duplicadas con el fin de mejorar la calidad del conjunto de datos.

Tabla 10
Ejemplos de registros eliminados durante la limpieza

spa	quz	Motivo de eliminación
””	”algo”	Valor vacío en spa
”Hola”	””	Valor vacío en quz
” ”	”algo”	Solo espacios en spa

Nota. Ejemplo de registros eliminados donde alguno de los dos lados esté vacío o sea solo espacios.

Figura 10
Algoritmo para eliminar pares vacíos y pares con espacios

```
mask = (
    (df["spa"] == "")
    |
    (df["quz"] == "")
)

df = df[~mask].copy()

df = df.drop_duplicates(
    subset=["spa", "quz"]
)
```

Seguidamente, se aplicó un proceso de separación de signos de puntuación, debido a que NLLB-200 utiliza un tokenizer basado en subpalabras (*SentencePiece*), el cual requiere una segmentación adecuada del texto para mejorar la tokenización. Este procedimiento resulta especialmente útil en idiomas aglutinantes como el quechua collao. Sin embargo, se tuvo especial cuidado con los apóstrofes propios del idioma, evitando separarlos cuando forman parte de una palabra.

Tabla 11
Ejemplos de separación de signos de puntuación

Texto original	Resultado procesado
¿Hayk'a qolqetan suwakuranku?	¿ Hayk'a qolqetan suwakuranku ?
¡Oye niño! Corre llama a tu papá	¡ Oye niño ! Corre llama a tu papá
Kaypiqa nishutan chirin	Kaypiqa nishutan chirin

Nota. Ejemplo de separación de los signos de puntuación, manteniendo intactos los apóstrofes internos del quechua.

Figura 11
Algoritmo para la separación de signos de puntuación

```
PUNCT_APERTURA = re.compile(r'([\¿\¡([{"«])')
PUNCT_CIERRE = re.compile(r'([.,?!)]}"»:;])')

def separar_puntuacion(texto: str) -> str:
    texto = PUNCT_APERTURA.sub(r' \1 ', texto)
    texto = PUNCT_CIERRE.sub(r' \1 ', texto)
    texto = re.sub(r' {2,}', ' ', texto)
    return texto.strip()

df["spa"]  = df["spa"].progress_apply(separar_puntuacion)
df["quz"] = df["quz"].progress_apply(separar_puntuacion)
```

Nota. El procedimiento inserta espacios alrededor de los signos de puntuación para facilitar la tokenización mediante SentencePiece en el modelo NLLB-200.

Después de finalizar el proceso de limpieza, se generaron versiones intermedias del corpus limpio con el fin de evitar la ejecución completa del pipeline en futuras ejecuciones.

4.5.3 Tokenización y filtrado en preprocesamiento

En esta etapa se realizó la tokenización con el objetivo de medir la longitud de cada oración en tokens. Esto permitió identificar aquellos pares de oraciones que superaban los 150 tokens en cualquiera de los dos idiomas, los cuales fueron eliminados posteriormente, ya que incrementaban considerablemente el consumo de memoria GPU durante el entrenamiento.

El límite de 150 tokens fue seleccionado como un balance entre la cobertura del corpus y la eficiencia computacional durante el *fine-tuning*. Aunque el modelo NLLB soporta secuencias más largas, trabajar con textos extensos en GPUs de capacidad moderada reduce

significativamente el tamaño del *batch* y aumenta el tiempo de entrenamiento.

Además, se aplicó un segundo filtro basado en la proporción de longitud entre ambas oraciones. Se eliminó cualquier par cuya diferencia superara una relación de 3:1 entre la cantidad de tokens, debido a que este tipo de casos suele indicar errores de alineación en el corpus, como frases incompletas, texto adicional o segmentaciones incorrectas. Este tipo de ruido afecta negativamente el aprendizaje de correspondencia entre ambos idiomas.

Tabla 12
Ejemplo ilustrativo de un par descartado por diferencia excesiva de longitud

Idioma	**Oración**	**Tokens**
Español	“el presidente dijo”	8
Quechua	“chay umalliq nisqa ...”	31

Nota. Las oraciones y cantidades de tokens se muestran únicamente con fines ilustrativos y no corresponde a un caso específico del corpus.

La Tabla 12 presenta un ejemplo de un par de traducción descartado por exceder el límite de proporción de longitud establecido. En este caso, la relación obtenida fue:

$$\frac{31}{8} = 3.87 > 3.0$$

Dado que el valor supera el umbral definido, el par fue eliminado durante la etapa de filtrado.

En este punto del preprocesamiento, se almacenó el corpus filtrado en un archivo JSONL con el fin de evitar la reejecución de todo el pipeline de limpieza y filtrado. Este formato resulta idóneo para el manejo de texto bilingüe, ya que preserva adecuadamente caracteres especiales, comillas y delimitadores dentro de las oraciones.

Figura 12
Almacenamiento del corpus filtrado en formato JSONL

```
CHECKPOINT_2 = os.path.join(
    DIR_DATA,
    "CheckPoint_2_Corpus_Filtrado.jsonl"
)

# Guardar únicamente las columnas de texto
df[["spa", "quz"]].to_json(
    CHECKPOINT_2,
    orient="records",
    lines=True,
    force_ascii=False
)
```

Nota. Exportación del corpus paralelo filtrado en formato JSONL para su reutilización en etapas posteriores del pipeline.

4.5.4 División del corpus en conjuntos de entrenamiento, validación y prueba

Para obtener mejores resultados durante el entrenamiento del modelo, el corpus paralelo fue dividido en tres subconjuntos como se detalla en la Tabla 4: conjunto de entrenamiento (*train*), conjunto de validación (*validation*) y conjunto de prueba (*test*). Esta división permite entrenar el modelo, monitorear su desempeño durante el entrenamiento y finalmente realizar una evaluación con datos no vistos previamente.

En esta etapa, el corpus todavía se encuentra en un *DataFrame* de pandas; sin embargo, posteriormente es convertido a un *Dataset* de Hugging Face, debido a que la clase *Trainer* utilizada durante el *fine-tuning* trabaja de manera más eficiente con este formato. Además, los conjuntos de datos fueron almacenados en formato Apache Arrow, ya que este permite una carga más rápida en futuros experimentos y evita repetir el proceso de partición.

Para asegurar la reproducibilidad de los experimentos, se utilizó una semilla fija $seed = 42$ durante la división de los datos. La separación del corpus se realizó mediante la función *train_test_split()* de *scikit-learn*, tal como se muestra en el siguiente fragmento de código.

Figura 13
Algoritmo para la división del corpus paralelo

```python
from sklearn.model_selection import train_test_split

SPLIT_VAL  = 0.10
SPLIT_TEST = 0.10
SEED = 42

# Separar conjunto test
df_temp, df_test = train_test_split(
    df_data,
    test_size=SPLIT_TEST,
    random_state=SEED
    )

# Separar train y validation
df_train, df_val = train_test_split(
    df_temp,
    test_size=SPLIT_VAL / (1 - SPLIT_TEST),
    random_state=SEED
    )
```

Nota. División del corpus paralelo en conjuntos de entrenamiento, validación y prueba.

Posteriormente, cada subconjunto fue convertido a formato *Dataset* compatible con Hugging Face.

Figura 14
Conversión de subconjuntos a formato Dataset de Hugging Face

```python
from datasets import Dataset

ds_train = Dataset.from_pandas(
    df_train.reset_index(drop=True)
    )

ds_val = Dataset.from_pandas(
    df_val.reset_index(drop=True)
    )

ds_test = Dataset.from_pandas(
    df_test.reset_index(drop=True)
    )
```

Finalmente, los conjuntos de datos fueron almacenados en formato Arrow para reutilizarlos posteriormente sin necesidad de repetir nuevamente el proceso de división.

Figura 15
Almacenamiento del conjunto de datos en formato Arrow

```
CHECKPOINT_3_TRAIN = os.path.join(
    DIR_DATA,
    "CheckPoint_3_HF_Train"
    )

ds_train.save_to_disk(
    CHECKPOINT_3_TRAIN
    )

# ...
```

Nota. Almacenamiento del conjunto de datos de entrenamiento en formato Arrow compatible con Hugging Face.

4.5.5 Tokenización

Una vez completada la limpieza y división del corpus paralelo, se realizó la etapa de tokenización. Este proceso consiste en convertir las oraciones en pequeñas unidades llamadas *tokens*, las cuales pueden ser palabras completas o fragmentos de palabras. Posteriormente, estos tokens son transformados en valores numéricos para que puedan ser procesados por el modelo durante el entrenamiento.

En esta investigación se utilizó el tokenizador *NllbTokenizer*, dado que el modelo NLLB fue entrenado originalmente con su propio tokenizador basado en SentencePiece. Este tokenizador cuenta con un vocabulario multilingüe de aproximadamente 256 mil tokens, permitiendo trabajar con distintos idiomas, incluyendo lenguas de bajos recursos.

Un detalle importante al configurar NLLB-200 es que no incluye una etiqueta propia para el quechua collao (*quz_Latn*), sino únicamente para el quechua ayacuchano (*quy_Latn*). Como ambas variantes pertenecen al quechua sureño y comparten gran parte de su vocabulario y estructura gramatical, se utilizó *quy_Latn* como idioma de destino (*target language*). De esta forma, el modelo aprovecha los tokens y representaciones ya aprendidas para adaptarse a la variante quechua collao durante el *fine-tuning*.

Asimismo, cada subconjunto del corpus (*train*, *validation* y *test*) fue tokenizado por

separado. Esto permitió mantener una organización modular de los datos y reutilizar posteriormente los datasets tokenizados sin necesidad de repetir nuevamente el proceso.

Durante la tokenización se definió un máximo de 150 tokens tanto para las oraciones de entrada como de salida, manteniendo consistencia con los filtros aplicados previamente durante la limpieza del corpus.

Figura 16
Configuración e inicialización del tokenizador NLLB-200

```
from transformers import NllbTokenizer

MODEL_NAME = "facebook/nllb-200-distilled-600M"

SRC_LANG = "spa_Latn"
TGT_LANG = "quy_Latn"

MAX_INPUT_LENGTH  = 150
MAX_TARGET_LENGTH = 150

tokenizer = NllbTokenizer.from_pretrained(
    MODEL_NAME
    )

tokenizer.src_lang = SRC_LANG
tokenizer.tgt_lang = TGT_LANG
```

Nota. Configuración inicial del tokenizador NLLB-200 utilizado en la investigación.

4.5.6 Migración del pipeline hacia RunPod

Las primeras etapas del procesamiento se realizaron en Google Colab debido a su facilidad para trabajar con notebooks y realizar pruebas rápidas sobre el corpus paralelo. Sin embargo, al incrementarse el tamaño del dataset y las exigencias computacionales del entrenamiento, fue necesario migrar las etapas finales a RunPod, utilizando instancias GPU de mayor capacidad.

Para organizar mejor el experimento y facilitar su reproducibilidad, el pipeline se dividió en dos etapas: entrenamiento y evaluación. Esta separación permitió reutilizar los datos previamente procesados sin repetir las fases de limpieza y tokenización, además de facilitar la ejecución de distintos experimentos modificando únicamente los parámetros de

entrenamiento. La configuración experimental utilizada se resume en la Tabla 13.

La etapa de entrenamiento consistió en la carga del modelo, la aplicación de LoRA y el entrenamiento supervisado. Por su parte, la etapa de evaluación se encargó de generar traducciones automáticas y calcular las métricas de desempeño en el conjunto de prueba.

Tabla 13
Configuración de recursos computacionales y parámetros de entrenamiento

Recurso / Parámetro	**Descripción**
Plataforma inicial	Google Colab
Plataforma de entrenamiento	RunPod
GPU	NVIDIA RTX 6000 Ada Generation
Precisión	FP16
Modelo base	facebook/nllb-200-distilled-600M
Longitud máxima de secuencia	150 tokens (entrada y salida)
Batch size	32
Épocas	5
Learning rate	2e-4
Warmup steps	100
Weight decay	0.01
LoRA r	16 (variado en experimentos)
LoRA alpha	32 (variado en experimentos)

Nota. Los parámetros de entrenamiento (batch size, épocas, learning rate, precisión, longitud de secuencia y configuración base del modelo) se mantuvieron constantes en los experimentos realizados, variando únicamente los hiperparámetros específicos de LoRA (r y alpha) y la proporción del conjunto de datos empleado. Esta estrategia permitió una comparación controlada del desempeño del modelo.

4.5.7 Inicialización del modelo e implementación de LoRA

Después de preparar el corpus paralelo, se inicializó el modelo base *NLLB-200-distilled-600M* utilizando la biblioteca Transformers.

Figura 17
Carga del modelo base NLLB-200 preentrenado

```
from transformers import AutoModelForSeq2SeqLM

modelo_base = AutoModelForSeq2SeqLM.from_pretrained(
    "facebook/nllb-200-distilled-600M"
)
```

Posteriormente, se implementó la técnica LoRA mediante la biblioteca PEFT, con el objetivo de reducir la cantidad de parámetros entrenables durante el *fine-tuning* del modelo. Esta técnica permite insertar adaptadores de bajo rango en determinadas capas del modelo, evitando modificar todos los parámetros originales y disminuyendo considerablemente el consumo de memoria GPU.

En esta investigación, LoRA fue aplicado sobre las capas de proyección *q_proj* y *v_proj*, las cuales forman parte del mecanismo de atención del modelo NLLB-200. A continuación, se presenta la configuración del valor de rango $r = 16$ y un factor de escalamiento $\alpha = 16$.

Figura 18
Configuración LoRA para el fine-tuning del modelo NLLB-200

```
from peft import LoraConfig, TaskType

lora_config = LoraConfig(
    task_type=TaskType.SEQ_2_SEQ_LM,
    r=16,
    lora_alpha=16,
    lora_dropout=0.1,
    target_modules=[
        "q_proj",
        "v_proj"
    ],
    bias="none"
)
```

Nota. Ejemplo de configuración de adaptadores LoRA utilizada durante el *fine-tuning* del modelo NLLB-200.

Finalmente, los adaptadores LoRA fueron insertados sobre el modelo base utilizando la función *get_peft_model()*, permitiendo entrenar únicamente una pequeña fracción de parámetros en comparación con el modelo completo.

Figura 19
Inserción de adaptadores LoRA sobre el modelo base

```
from peft import get_peft_model

modelo_lora = get_peft_model(
    modelo_base,
    lora_config
)
```

4.5.8 Configuración y entrenamiento del modelo

Después de implementar la técnica LoRA, se definieron los hiperparámetros utilizados durante el proceso de *fine-tuning* del modelo. Entre los principales parámetros configurados se encuentran la tasa de aprendizaje (*learning rate*), el tamaño de lote (*batch size*), la cantidad de épocas y el uso de precisión mixta (*fp16*) para reducir el consumo de memoria GPU durante el entrenamiento.

Asimismo, se utilizó *Seq2SeqTrainingArguments* de Hugging Face para centralizar la configuración del entrenamiento, evaluación y almacenamiento de checkpoints del modelo.

Figura 20
Configuración de hiperparámetros de entrenamiento

```
training_args = Seq2SeqTrainingArguments(
    # ...
    num_train_epochs=5,
    per_device_train_batch_size=32,
    learning_rate=2e-4,
    # ...
    eval_strategy="epoch",
    fp16=True,
    seed=42,
    # ...
)
```

Nota. Fragmento de la configuración utilizada durante el entrenamiento del modelo.

Posteriormente, se utilizó *DataCollatorForSeq2Seq* para organizar dinámicamente los lotes de entrenamiento y aplicar correctamente el *padding* en las secuencias de entrada y salida.

Figura 21
Configuración del DataCollator para los lotes de entrenamiento

```
data_collator = DataCollatorForSeq2Seq(
    tokenizer=tokenizer,
    model=modelo_lora,
    padding=True,
    label_pad_token_id=-100
)
```

Nota. Configuración utilizada para organizar los lotes y aplicar *padding* durante el entrenamiento.

Finalmente, el entrenamiento fue ejecutado mediante *Seq2SeqTrainer*, utilizando el conjunto de datos de entrenamiento para el aprendizaje del modelo y el conjunto de datos de validación para monitorear el desempeño durante cada época.

Figura 22
Ejecución del entrenamiento mediante Seq2SeqTrainer

```
trainer = Seq2SeqTrainer(
    model=modelo_lora,
    args=training_args,
    train_dataset=ds_train_tok,
    eval_dataset=ds_val_tok,
    data_collator=data_collator
)

trainer.train()
```

Nota. Inicialización y ejecución del entrenamiento utilizando *Seq2SeqTrainer*.

Durante el entrenamiento se monitoreó constantemente el uso de memoria GPU, la pérdida de entrenamiento y la pérdida de validación, tal como se muestra en la Tabla 14.

Durante las pruebas iniciales se realizaron distintos experimentos modificando parámetros como *batch size*, *learning rate* y número de épocas hasta encontrar una configuración estable para el entrenamiento.

4.5.9 Evaluación del modelo

Finalmente, el modelo ajustado fue evaluado utilizando el conjunto de datos de prueba, el cual no fue utilizado durante el entrenamiento. Para ello, primero se generaron traducciones automáticas a partir de las oraciones en español y posteriormente estas fueron

Tabla 14
Evolución de pérdida durante el entrenamiento

Época	Train Loss	Validation Loss
1	2.461373	2.242182
2	2.340375	2.143084
3	2.268417	2.088454
4	2.287892	2.065521
5	2.265876	2.055581

Nota. Evolución de la pérdida de entrenamiento y validación durante el proceso de *fine-tuning*. La configuración del experimento correspondió a LoRA con $r = 16$ y $\alpha = 32$.

comparadas con las traducciones de referencia en quechua collao.

Durante la generación de traducciones se utilizó el método *generate()* del modelo NLLB junto con búsqueda por haces (*beam search*) para mejorar la calidad de las predicciones generadas.

Figura 23
Generación de traducciones automáticas

```
outputs = modelo_lora.generate(
    **inputs,
    forced_bos_token_id=tgt_lang_id,
    num_beams=4

    #...
)
```

Nota. Generación de traducciones automáticas utilizando el modelo ajustado.

Posteriormente, las traducciones generadas fueron evaluadas utilizando las métricas automáticas comúnmente utilizadas en TAN, específicamente BLEU, ChrF y ChrF++ mediante la biblioteca *evaluate*.

Figura 24
Cálculo de métricas de evaluación automática (BLEU, ChrF y ChrF++)

```
bleu_result = bleu_metric.compute(
    predictions=predicciones,
    references=referencias_qu
)

chrf_result = chrf_metric.compute(
    predictions=predicciones,
    references=referencias_qu
)

chrfpp_result = chrf_metric.compute(
    predictions=predicciones,
    references=referencias_qu,
    word_order=2
)
```

Nota. Evaluación automática del modelo mediante las métricas BLEU, ChrF y ChrF++.

4.6 Consideraciones éticas

La presente investigación fue desarrollada con fines estrictamente académicos y de investigación científica. Los corpus utilizados durante el entrenamiento fueron obtenidos desde fuentes abiertas y repositorios públicos utilizados frecuentemente en investigaciones relacionadas con PLN y TAN.

Durante el desarrollo del experimento se procuró respetar las referencias correspondientes de cada dataset utilizado, así como las licencias y condiciones de uso publicadas por sus autores o instituciones responsables.

Asimismo, se consideró importante mantener respeto por la representación lingüística y cultural del quechua collao, debido a que se trata de una lengua originaria perteneciente al patrimonio cultural del Perú. En este sentido, la investigación no tuvo como objetivo reemplazar el trabajo humano de traducción ni establecer una forma correcta única del idioma, sino explorar el uso de técnicas de IA orientadas al fortalecimiento tecnológico de lenguas de bajos recursos.

También se reconoce que los sistemas de TAN presentan limitaciones inherentes rela-

cionadas con errores semánticos, ambigüedad contextual y variaciones dialectales. Debido a ello, las traducciones generadas por el modelo deben ser interpretadas como resultados experimentales y no como sustitutos definitivos de traducciones realizadas por hablantes expertos o especialistas lingüísticos.

De igual manera, se procuró mantener transparencia metodológica durante todas las etapas del experimento. Para ello, se documentaron los principales procedimientos relacionados con limpieza del corpus, tokenización, entrenamiento y evaluación del modelo.

Finalmente, la investigación buscó contribuir al desarrollo de recursos tecnológicos relacionados con lenguas originarias y fomentar futuras investigaciones en el área de TAN aplicada a la lengua quechua collao.

V. Resultados y discusión

5.1 Resultados de la adaptación del modelo NLLB-200

Con el propósito de adaptar el modelo *NLLB-200-distilled-600M* para la traducción automática español–quechua collao, se realizaron experimentos de *fine-tuning* mediante la técnica LoRA. Estos se organizaron en tres grupos de comparación para analizar el efecto de la cantidad de corpus paralelo, el rango (r) y el factor de escala (α), evaluando tres configuraciones en cada caso.

Tabla 15
Resultados de los grupos evaluados

Grupo	Exp	Corpus	r	α	BLEU	ChrF	ChrF++
Tamaño del corpus	Exp 1	25 %	16	16	8.84	38.61	34.05
	Exp 2	50 %	16	16	8.92	39.18	34.58
	Exp 3	100 %	16	16	9.19	39.90	35.22
Parámetro r	Exp 4	100 %	8	16	9.20	39.76	35.11
	Exp 5	100 %	16	16	9.19	39.90	35.22
	Exp 6	100 %	32	16	9.22	39.87	35.21
Parámetro α	Exp 7	100 %	16	8	8.92	39.25	34.62
	Exp 8	100 %	16	16	9.19	39.90	35.22
	Exp 9	100 %	16	32	9.50	40.52	35.81

Nota. Los experimentos 3, 5 y 8 corresponden a la misma configuración base (corpus = 100 %, $r = 16$ y $\alpha = 16$).

Tabla 16
Tiempo de entrenamiento y uso de memoria GPU según la configuración evaluada

Grupo	Configuración	Tiempo de entrenamiento	Memoria GPU (GB)
Tamaño del corpus	25 %	483.20 s (8.05 min)	22.20
	50 %	966.46 s (16.11 min)	22.19
	100 %	1804.71 s (30.08 min)	22.20
Parámetro r	8	1868.64 s (31.14 min)	22.18
	32	1881.22 s (31.35 min)	22.24
Parámetro α	8	1769.43 s (29.49 min)	22.20
	32	2401.32 s (40.02 min)	22.20

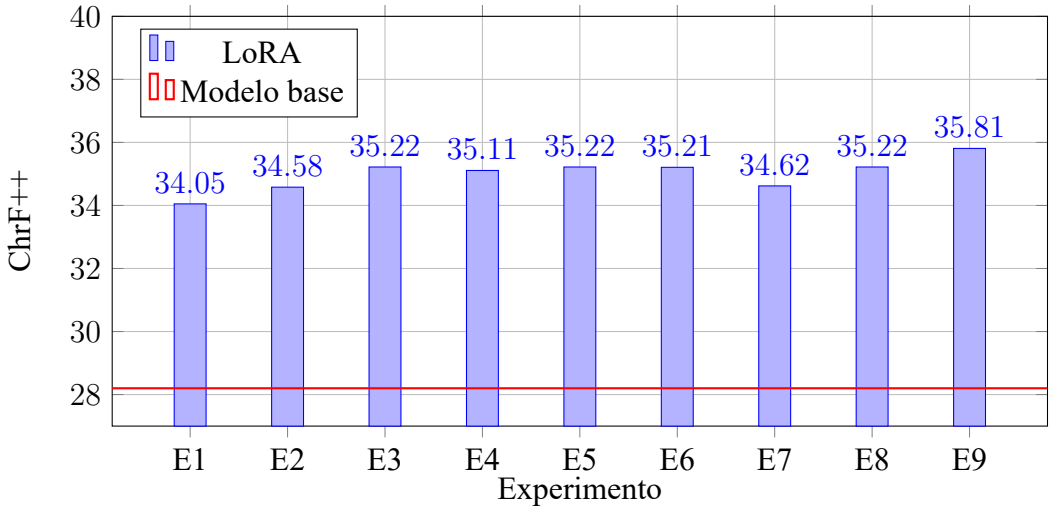
Nota. En la evaluación del tamaño del corpus se mantuvieron constantes $r = 16$ y $\alpha = 16$. En la evaluación del parámetro r se utilizó el 100 % del corpus y $\alpha = 16$. En la evaluación del parámetro α se utilizó el 100 % del corpus y $r = 16$.

Tabla 17
Resumen de eficiencia computacional del mejor modelo

Modelo	BLEU	ChrF	ChrF++	Tiempo (min)	GPU (GB)
Base NLLB-200	4.82	32.99	28.20	–	–
LoRA	9.50	40.52	35.81	30.08	22.20

Nota. Se compara el desempeño del modelo base NLLB-200 con el mejor modelo obtenido mediante *fine-tuning* con la técnica LoRA. Los valores de tiempo y memoria corresponden al proceso de entrenamiento de este último.

Figura 25
Comparación de configuraciones LoRA según ChrF++



Nota. La línea horizontal representa el desempeño del modelo base NLLB-200 (ChrF++ = 28.20). El mejor resultado corresponde al experimento 9, con ChrF++ = 35.81.

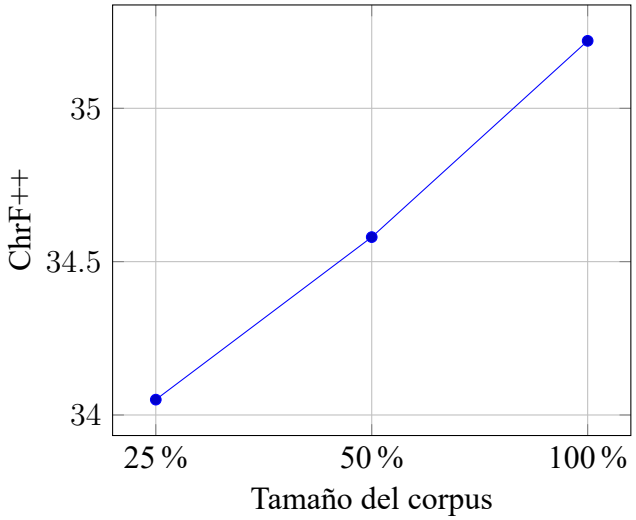
5.1.1 Influencia del tamaño del corpus

Tabla 18
Resultados del Grupo A: variación del tamaño del corpus

Experimento	Corpus	r	α	BLEU	ChrF	ChrF++
Exp. 1	25 %	16	16	8.84	38.61	34.05
Exp. 2	50 %	16	16	8.92	39.18	34.58
Exp. 3	100 %	16	16	9.19	39.90	35.22

Los resultados muestran una tendencia creciente conforme aumenta la cantidad de corpus utilizado durante el *fine-tuning*. El experimento realizado con el 100 % del corpus obtuvo los mejores resultados, alcanzando un BLEU de 9.19, un ChrF de 39.90 y un ChrF++ de 35.22. En contraste, el modelo entrenado con únicamente el 25 % de los datos obtuvo un BLEU de 8.84, un ChrF de 38.61 y un ChrF++ de 34.05.

Figura 26
Efecto del tamaño del corpus en el desempeño del modelo



Estos resultados sugieren que el incremento del volumen de datos paralelos contribuye positivamente al aprendizaje de patrones lingüísticos entre el español y el quechua collao, permitiendo mejorar la calidad de las traducciones generadas.

5.1.2 Influencia del parámetro r

Tabla 19
Resultados del Grupo B: variación del parámetro r

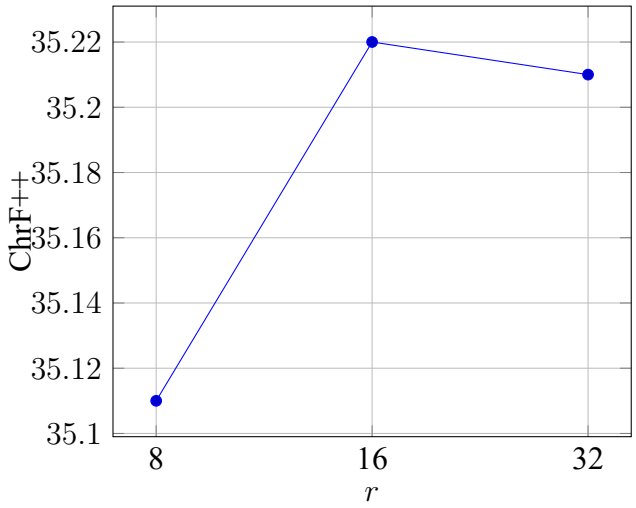
Experimento	Corpus	r	α	BLEU	ChrF	ChrF++
Exp. 4	100 %	8	16	9.20	39.76	35.11
Exp. 5	100 %	16	16	9.19	39.90	35.22
Exp. 6	100 %	32	16	9.22	39.87	35.21

Los resultados evidencian diferencias mínimas entre los valores evaluados para el parámetro r. El mejor desempeño se obtuvo en el experimento 5 ($r = 16$), con un puntaje de 9.19 en BLEU, 39.90 en ChrF y 35.22 en ChrF++. No obstante, las variaciones respecto a los experimentos con $r = 8$ y $r = 32$ son muy reducidas, con diferencias menores a 0.1 puntos en todas las métricas.

Estos resultados indican que, para el corpus utilizado, el incremento del rango de

adaptación no genera mejoras significativas en la calidad de la traducción, lo que sugiere que valores moderados de r son suficientes para este escenario experimental.

Figura 27
Impacto del parámetro r en el rendimiento del modelo



5.1.3 Influencia del factor de escala α

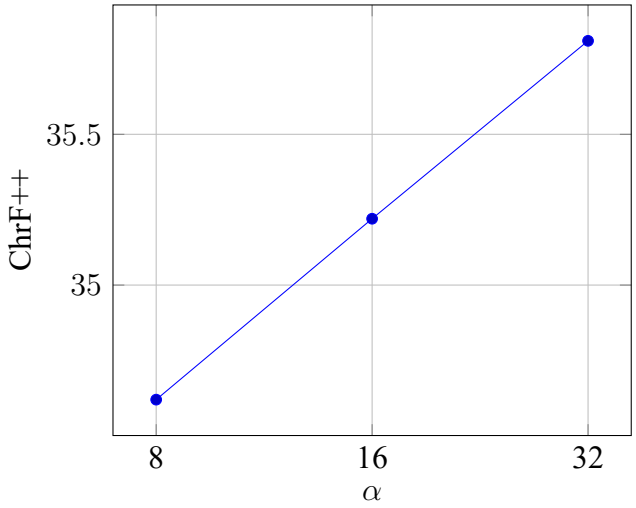
Tabla 20
Resultados del Grupo C: variación del parámetro α

Experimento	Corpus	r	α	BLEU	ChrF	ChrF++
Exp. 7	100 %	16	8	8.92	39.25	34.62
Exp. 8	100 %	16	16	9.19	39.90	35.22
Exp. 9	100 %	16	32	9.50	40.52	35.81

Los resultados muestran que el parámetro α tiene una influencia más notable en comparación con el parámetro r. El mejor desempeño se obtuvo en el experimento 9 ($\alpha = 32$), alcanzando un BLEU de 9.50, ChrF de 40.52 y ChrF++ de 35.81, que corresponden a los valores más altos del estudio.

Asimismo, se observa una mejora progresiva en las métricas a medida que aumenta el valor de α, especialmente entre los experimentos con $\alpha = 8$ y $\alpha = 32$. Esto sugiere que un mayor factor de escala podría favorecer la adaptación del modelo al dominio lingüístico del corpus utilizado.

Figura 28
Efecto del factor α en la calidad de traducción



5.2 Comparación entre el modelo base y el modelo adaptado

Con el objetivo de determinar el impacto del *fine-tuning* mediante la técnica LoRA, se comparó el desempeño del mejor experimento obtenido con respecto al modelo base original *NLLB-200-distilled-600M*.

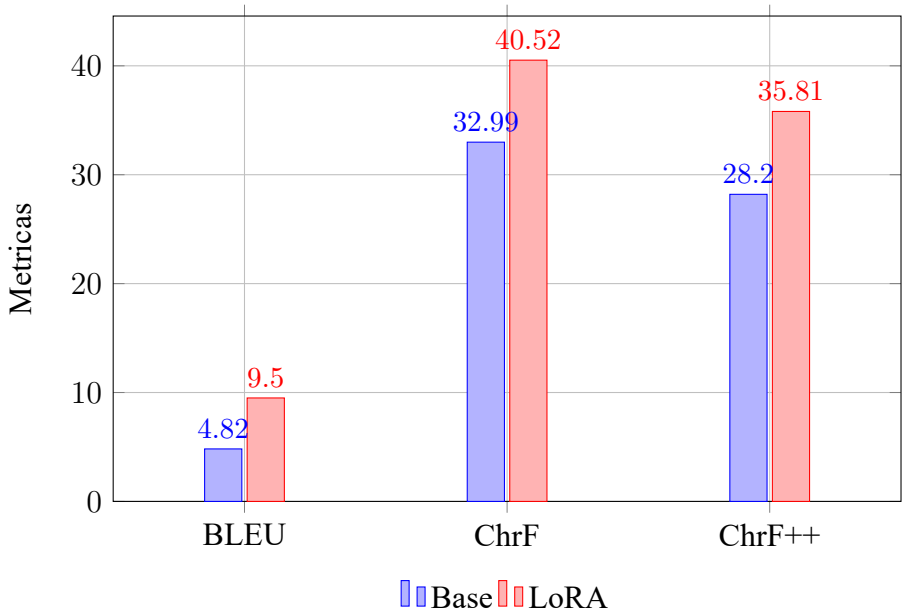
Tabla 21
Comparación entre el modelo base y el mejor modelo adaptado

Modelo	**BLEU**	**ChrF**	**ChrF++**
Modelo Base	4.82	32.99	28.20
Modelo Adaptado (Exp. 9)	9.50	40.52	35.81
Diferencia	**+4.68**	**+7.53**	**+7.61**

Los resultados muestran que el *fine-tuning* mediante la técnica LoRA permitió incrementar todas las métricas de evaluación. El BLEU prácticamente se duplicó respecto al modelo base, mientras que ChrF y ChrF++ registraron incrementos superiores a siete puntos.

Estos hallazgos evidencian que la adaptación específica al dominio español–quechua collao permitió mejorar la calidad de las traducciones generadas por el modelo.

Figura 29
Comparación del modelo base vs modelo adaptado



Nota. Se compara el modelo base *NLLB-200-distilled-600M* con el mejor modelo obtenido mediante *fine-tuning* con la técnica LoRA. La configuración seleccionada corresponde al experimento 9, que alcanzó los valores más altos de BLEU, ChrF y ChrF++.

5.3 Discusiones

El objetivo general de la presente investigación fue determinar el efecto de la adaptación del modelo NLLB-200 en la TA de la lengua quechua collao en la región de Cusco. Los resultados obtenidos evidencian que la adaptación del modelo mediante la técnica LoRA tuvo un efecto positivo, reflejado en la mejora de las métricas de evaluación en comparación con el modelo base.

Asimismo, los experimentos realizados demuestran que las distintas configuraciones evaluadas influyen en el desempeño del modelo adaptado, observándose variaciones moderadas en las métricas BLEU, ChrF y ChrF++, así como una tendencia general de mejora cuando se optimizan los parámetros de adaptación y el aprovechamiento del corpus de entrenamiento.

Estos resultados son consistentes con lo reportado por Gow-Smith y Sánchez Villegas (2023) y Ahmed et al. (2023), quienes señalan que la adaptación de modelos multilingües

mediante corpus paralelos específicos mejora la calidad de traducción en lenguas de bajos recursos. De manera similar, la presente investigación evidencia que la especialización del modelo NLLB-200 permite obtener mejores resultados para el par español–quechua collao.

En términos teóricos, este comportamiento responde a que el proceso de *fine-tuning* permite orientar el conocimiento previamente aprendido por el modelo hacia las particularidades del quechua collao, conservando al mismo tiempo la información adquirida durante el preentrenamiento.

No obstante, aunque la adaptación permitió mejorar el desempeño general del sistema, aún se observaron diferencias entre algunas traducciones generadas y las traducciones de referencia, lo que evidencia que existe margen de mejora. Estas diferencias pueden estar asociadas a la limitada disponibilidad de corpus paralelos y a la complejidad morfológica y sintáctica propia del quechua collao.

Respecto al primer objetivo específico, orientado a aplicar la técnica LoRA para adaptar el modelo NLLB-200, los resultados evidencian que la técnica fue implementada satisfactoriamente durante el proceso de *fine-tuning*, permitiendo adaptar el modelo sin necesidad de actualizar la totalidad de sus parámetros.

Este resultado coincide con lo reportado por Garcia Gilabert et al. (2024) y Liang et al. (2025), quienes sostienen que LoRA constituye una alternativa eficiente para la adaptación de modelos de gran tamaño, manteniendo un desempeño comparable al *fine-tuning* completo con un número considerablemente menor de parámetros entrenables.

El fundamento de este resultado radica en la propia arquitectura de LoRA, la cual incorpora matrices de bajo rango que permiten aprender nueva información sin modificar directamente los parámetros originales del modelo NLLB-200, reduciendo así el costo computacional del entrenamiento.

Asimismo, durante los experimentos se observó que el consumo de memoria GPU permaneció prácticamente constante, alrededor de 22.2 GB, independientemente de la con-

figuración utilizada. Esto demuestra la eficiencia computacional de la técnica. Sin embargo, el entrenamiento continúa requiriendo hardware especializado debido al tamaño del modelo base, lo que puede representar una limitación para reproducir el estudio en equipos con menores capacidades de procesamiento.

En relación con el segundo objetivo específico, referido a evaluar el rendimiento del modelo adaptado mediante diferentes métricas de TA, los resultados obtenidos evidencian variaciones en el desempeño del modelo en función de las configuraciones evaluadas. En particular, se observa que el parámetro α presenta una mayor influencia en el rendimiento del modelo en comparación con el parámetro r, reflejando mejoras más consistentes en las métricas BLEU, ChrF y ChrF++.

Estos hallazgos guardan relación con lo señalado por Liang et al. (2025), quienes indican que una adecuada configuración de los hiperparámetros durante el proceso de adaptación permite obtener mejoras significativas en tareas de traducción para lenguas de bajos recursos.

Dicho comportamiento encuentra sustento en el rol de cada hiperparámetro, debido a que el parámetro α controla la intensidad con la que las actualizaciones aprendidas por LoRA son incorporadas al modelo, mientras que el parámetro r determina la dimensión de las matrices de adaptación. En consecuencia, una adecuada combinación de ambos parámetros favorece el aprendizaje durante el proceso de *fine-tuning*.

Por otra parte, el análisis de las métricas permitió observar diferencias en la forma en que cada una evalúa la calidad de la traducción. Mientras que BLEU realiza una comparación basada principalmente en coincidencias de palabras, ChrF++ considera coincidencias a nivel de caracteres y subpalabras, aspecto que resulta particularmente útil para idiomas con alta riqueza morfológica como el quechua. Esto explica por qué algunas traducciones obtuvieron diferencias entre ambas métricas aun cuando conservaron adecuadamente el significado de la oración.

Finalmente, respecto al tercer objetivo específico, los resultados permitieron com-

parar el desempeño del modelo NLLB-200 adaptado frente al modelo base, evidenciando mejoras en todas las métricas de evaluación consideradas. Estos resultados permiten aceptar la hipótesis específica correspondiente, ya que la adaptación mediante la técnica LoRA incrementó el desempeño del sistema de TA.

Los resultados obtenidos son consistentes con lo reportado por Chen et al. (2024), quienes demostraron que la especialización de modelos multilingües mediante técnicas de *fine-tuning* mejora la calidad de traducción en lenguas de bajos recursos. De igual manera, la presente investigación confirma que la adaptación de un modelo previamente entrenado resulta más efectiva que utilizar únicamente el modelo base sin especialización.

Teóricamente, esta mejora se debe a que el proceso de adaptación permite que el modelo aprenda patrones lingüísticos específicos del quechua collao que no se encuentran suficientemente representados durante el preentrenamiento general del modelo.

A pesar de las mejoras alcanzadas, la comparación también permitió identificar que aún existen diferencias entre las traducciones generadas por el modelo adaptado y las traducciones de referencia, especialmente en construcciones lingüísticas de mayor complejidad. Esto indica que, aunque la adaptación redujo las limitaciones presentes en el modelo base, todavía existe margen para continuar mejorando el desempeño mediante corpus paralelos de mayor tamaño y futuras estrategias de optimización.

VI. Conclusiones

Se determinó que la adaptación del modelo *NLLB-200-distilled-600M* mediante la técnica LoRA tuvo un efecto positivo en la traducción automática español–quechua collao. Los resultados evidenciaron mejoras en todas las métricas de evaluación respecto al modelo base y permitieron observar el impacto de distintas configuraciones de *fine-tuning* sobre el desempeño del sistema. En particular, el tamaño del corpus y el parámetro α mostraron una influencia favorable en la calidad de las traducciones obtenidas. En consecuencia, se concluye que la adaptación de modelos multilingües mediante la técnica LoRA representa una alternativa viable para mejorar sistemas de traducción automática en lenguas de bajos recursos como el quechua collao.

- La aplicación de la técnica LoRA permitió adaptar satisfactoriamente el modelo *NLLB-200-distilled-600M* para la traducción automática español–quechua collao. El proceso de *fine-tuning* se realizó de manera eficiente, manteniendo un consumo estable de memoria GPU cercano a 22.2 GB durante los experimentos, lo cual confirma la eficiencia computacional del enfoque propuesto y su viabilidad en infraestructuras accesibles.
- La evaluación del modelo adaptado mediante métricas automáticas demostró un desempeño estable en las diferentes configuraciones analizadas. El mejor modelo alcanzó valores de 9.50 en BLEU, 40.52 en ChrF y 35.81 en ChrF++, evidenciando una mejora respecto a las demás configuraciones evaluadas. Asimismo, se observó que el incremento del tamaño del corpus y el ajuste del parámetro α influyeron de manera favorable en el rendimiento del modelo.
- La comparación entre el modelo base y el mejor modelo adaptado confirmó la efectividad del proceso de adaptación mediante la técnica LoRA. En relación con el modelo

original, se obtuvieron mejoras de 4.68 puntos en BLEU, 7.53 puntos en ChrF y 7.61 puntos en ChrF++, lo que evidencia un incremento en la calidad de las traducciones generadas para el par español–quechua collao.

VII. Recomendaciones

Se recomienda ampliar el corpus paralelo español–quechua collao incorporando fuentes más diversas como textos educativos, discursos orales transcritos, documentos administrativos y contenido cultural. Esto permitiría mejorar la capacidad de generalización del modelo y reducir posibles sesgos derivados de datos limitados o poco variados.

Se recomienda explorar variantes de mayor tamaño del modelo *NLLB-200-distilled-600M*, como *NLLB-200-1.3B* y *NLLB-200-3.3B*, además de evaluar otros modelos multilingües recientes. Asimismo, resulta pertinente comparar la técnica LoRA con otros enfoques PEFT, como QLoRA o adapters, con el fin de analizar diferencias en rendimiento, estabilidad y consumo de recursos computacionales.

Se recomienda complementar la evaluación automática con evaluación humana, especialmente en lo referente a la adecuación semántica y cultural de las traducciones en lengua quechua, debido a que métricas como BLEU, ChrF y ChrF++ no capturan completamente la riqueza lingüística de las lenguas originarias.

Se recomienda mejorar la calidad del corpus paralelo mediante procesos más rigurosos de normalización lingüística y estandarización ortográfica, considerando la variabilidad dialectal del quechua collao, lo cual impacta directamente en la calidad del entrenamiento del modelo.

Se recomienda desarrollar investigaciones futuras orientadas a la traducción automática bidireccional (español–quechua y quechua–español), lo cual ampliaría significativamente el alcance del sistema en aplicaciones educativas, institucionales y en iniciativas orientadas a la preservación de lenguas originarias.

Se recomienda mantener una arquitectura modular del pipeline experimental, separando claramente los procesos de preprocesamiento, entrenamiento y evaluación. Esto facilita

la reproducibilidad de los experimentos, la escalabilidad del sistema y la comparación de configuraciones alternativas del modelo.

VIII. Referencias

Agić, Ž., & Vulić, I. (2019). JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages. En A. Korhonen, D. Traum & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3204-3210). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1310

Ahmed, N., Flechas Manrique, N., & Petrović, A. (2023). Enhancing Spanish-Quechua Machine Translation with Pre-Trained Models and Diverse Data Sources: LCT-EHU at AmericasNLP Shared Task. En M. Mager, A. Ebrahimi, A. Oncevay, E. Rice, S. Rijhwani, A. Palmer & K. Kann (Eds.), *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)* (pp. 156-162). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.americasnlp-1.16

Anisuzzaman, D. M., Malins, J. G., Friedman, P. A., & Attia, Z. I. (2025). Fine-Tuning Large Language Models for Specialized Use Cases. *Mayo Clinic Proceedings: Digital Health*, *3*(1), 100184. https://doi.org/10.1016/j.mcpdig.2024.11.005

AWS Amazon. (s.f.). *¿Qué es un LLM?* Amazon Web Services, Inc. Consultado el 2 de diciembre de 2025, desde https://aws.amazon.com/es/what-is/large-language-model/

Badman, A., & Kosinski, M. (s.f.). *What is a Dataset?* IBM. Consultado el 28 de mayo de 2026, desde https://www.ibm.com/think/topics/dataset

Banerjee, S., & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. En J. Goldstein, A. Lavie, C.-Y. Lin & C. Voss (Eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (pp. 65-72). Association for Computational Linguistics. https://aclanthology.org/W05-0909/

Belcic, I., & Stryker, C. (s.f.). *¿Qué es un pipeline de aprendizaje automático?* IBM. Consultado el 6 de junio de 2026, desde https://www.ibm.com/mx-es/think/topics/machine-learning-pipeline

Cerron-Palomino, R. (2003). *Lingüística quechua*. Centro de Estudios Regionales Andinos Bartolomé de Las Casas. https://isbn.cloud/9789972691591/linguistica-quechua/

Chauhan, S., & Daniel, P. (2023). A Comprehensive Survey on Various Fully Automatic Machine Translation Evaluation Metrics. *Neural Processing Letters*, *55*(9), 12663-12717. https://doi.org/10.1007/s11063-022-10835-4

Chen, J., Shu, P., Li, Y., Zhao, H., Jiang, H., Pan, Y., Zhou, Y., Liu, Z., Howe, L. C., & Liu, T. (2024). *QueEn: A Large Language Model for Quechua-English Translation*. https://doi.org/10.48550/arXiv.2412.05184

Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A.,

Akula, B., Barrault, L., Gonzalez, G. M., Hansanti, P., Hoffman, J., … Wang, J. (2022). *No Language Left Behind: Scaling Human-Centered Machine Translation*. https://doi.org/10.48550/arXiv.2207.04672

DeGenaro, D., & Lupicki, T. (2024). Experiments in Mamba Sequence Modeling and NLLB-200 Fine-Tuning for Low Resource Multilingual Machine Translation. En M. Mager, A. Ebrahimi, S. Rijhwani, A. Oncevay, L. Chiruzzo, R. Pugh & K. von der Wense (Eds.), *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)* (pp. 188-194). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.americasnlp-1.22

Do Campo Bayón, M. (2023). *Traducción automática neuronal para lenguas con recursos reducidos. La evaluación de los usuarios según el principio de no inferioridad.* [Tesis doctoral, Universitat Autònoma de Barcelona]. https://www.tdx.cat/handle/10803/691201

Ebrahimi, A., de Gibert, O., Vazquez, R., Coto-Solano, R., Denisov, P., Pugh, R., Mager, M., Oncevay, A., Chiruzzo, L., von der Wense, K., & Rijhwani, S. (2024). Findings of the AmericasNLP 2024 Shared Task on Machine Translation into Indigenous Languages. En M. Mager, A. Ebrahimi, S. Rijhwani, A. Oncevay, L. Chiruzzo, R. Pugh & K. von der Wense (Eds.), *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)* (pp. 236-246). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.americasnlp-1.28

Garcia Gilabert, J., Sant, A., Escolano, C., De Luca Fornaciari, F., Mash, A., & Melero, M. (2024). BSC Submission to the AmericasNLP 2024 Shared Task. En M. Mager, A. Ebrahimi, S. Rijhwani, A. Oncevay, L. Chiruzzo, R. Pugh & K. von der Wense (Eds.), *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)* (pp. 143-149). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.americasnlp-1.17

Google Cloud. (s.f.). *Inteligencia artificial (IA): guía sencilla*. Google Cloud. Consultado el 2 de diciembre de 2025, desde https://cloud.google.com/learn/what-is-artificial-intelligence

Google Colab. (s.f.). *Google Colaboratory*. colab.google. Consultado el 12 de diciembre de 2025, desde https://colab.google

Google for Developers. (s.f.). *Glosario sobre aprendizaje automático*. Google for Developers. Consultado el 12 de diciembre de 2025, desde https://developers.google.com/machine-learning/glossary?hl=es-419

Gow-Smith, E., & Sánchez Villegas, D. (2023). Sheffield's Submission to the AmericasNLP Shared Task on Machine Translation into Indigenous Languages. En M. Mager, A. Ebrahimi, A. Oncevay, E. Rice, S. Rijhwani, A. Palmer & K. Kann (Eds.), *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of*

the Americas (AmericasNLP) (pp. 192-199). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.americasnlp-1.21

Guzman, A. C., & Avalos, F. J. (2025). *La autotraducción del quechua-castellano en el cine regional peruano: Un estudio de la gestión del proyecto de subtitulación de tres directores quechuahablantes* [Tesis de licenciatura]. Universidad Peruana de Ciencias Aplicadas. https://repositorioacademico.upc.edu.pe/handle/10757/684538

Haddow, B., Bawden, R., Miceli Barone, A. V., Helcl, J., & Birch, A. (2022). Survey of Low-Resource Machine Translation. *Computational Linguistics*, *48*(3), 673-732. https://doi.org/10.1162/coli_a_00446

Hernández-Sampieri, R., & Mendoza Torres, C. P. (2023). *Metodología de La Investigación: Las Rutas Cuantitativa, Cualitativa y Mixta*. McGraw-Hill Education. https://books.google.com.pe/books?id=xuGp0AEACAAJ

Horna-Saldaña, C. J., Perez Perez, J. E., & Toro Galeano, M. L. (2025). Artificial Intelligence in the Preservation of Native Languages and Bridging the Information Access Gap for Indigenous Peoples. *Journal of Enabling Technologies*, *19*(1), 63-75. https://doi.org/10.1108/JET-09-2024-0063

Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. https://doi.org/10.48550/arXiv.2106.09685

Hu, J. (2024). Neural Machine Translation (NMT): Deep Learning Approaches through Neural Network Models. *Applied and Computational Engineering*, *82*(1), 93-99. https://doi.org/10.54254/2755-2721/82/20240944

Huarcaya, D. (2020). *Traducción automática neuronal para lengua nativa peruana* [Tesis de bachiller]. Universidad Peruana Unión. https://hdl.handle.net/20.500.12840/4143

Hugging Face. (s.f.). *The AI Community Building the Future.* Hugging Face. Consultado el 12 de diciembre de 2025, desde https://huggingface.co/

Hus, J., Anastasopoulos, A., & Krasner, N. (2025). Machine Translation Using Grammar Materials for LLM Post-Correction. En M. Mager, A. Ebrahimi, R. Pugh, S. Rijhwani, K. Von Der Wense, L. Chiruzzo, R. Coto-Solano & A. Oncevay (Eds.), *Proceedings of the Fifth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)* (pp. 92-99). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.americasnlp-1.10

IBM. (s.f.). *¿Qué es el PLN?* IBM. Consultado el 2 de diciembre de 2025, desde https://www.ibm.com/es-es/think/topics/natural-language-processing

INEI. (2017). *Censos Nacionales 2017: XII de Población, VII de Vivienda y III de Comunidades Indígenas*. Consultado el 26 de noviembre de 2025, desde https://censo2017.inei.gob.pe/

Koehn, P. (2017). *Neural Machine Translation*. https://doi.org/10.48550/arXiv.1709.07809

Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., Koo, S., & Lim, H. (2023). A Survey on Evaluation Metrics for Machine Translation. *Mathematics*, *11*(4), 1006. https://doi.org/10.3390/math11041006

Lenovo. (s.f.). *¿Qué es una GPU?* Lenovo. Consultado el 12 de diciembre de 2025, desde https://www.lenovo.com/es/partssale/es/glossary/what-is-gpu/

Liang, X., Jasmina Khaw, Y.-M., Liew, S.-Y., Tan, T.-P., & Qin, D. (2025). Toward Low-Resource Languages Machine Translation: A Language-Specific Fine-Tuning With LoRA for Specialized Large Language Models. *13*, 46616-46626. https://doi.org/10.1109/ACCESS.2025.3549795

Liu, Y., He, H., Han, T., Zhang, X., Liu, M., Tian, J., Zhang, Y., Wang, J., Gao, X., Zhong, T., Pan, Y., Xu, S., Wu, Z., Liu, Z., Zhang, X., Zhang, S., Hu, X., Zhang, T., Qiang, N., … Ge, B. (2025). Understanding LLMs: A Comprehensive Overview from Training to Inference. *Neurocomputing*, *620*, 129190. https://doi.org/10.1016/j.neucom.2024.129190

Lu, W., Luu, R. K., & Buehler, M. J. (2025). Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, *11*(1), 84. https://doi.org/10.1038/s41524-025-01564-y

Mager, M., Bhatnagar, R., Neubig, G., Vu, N. T., & Kann, K. (2023). *Neural Machine Translation for the Indigenous Languages of the Americas: An Introduction*. https://doi.org/10.48550/arXiv.2306.06804

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2025). *Large Language Models: A Survey*. https://doi.org/10.48550/arXiv.2402.06196

Ministerio de Educación. (2018). *Lenguas Originarias del Perú*. https://centroderecursos.cultura.pe/es/registrobibliografico/lenguas-originarias-del-per%C3%BA

Mondal, S. K., Zhang, H., Kabir, H. M. D., Ni, K., & Dai, H.-N. (2023). Machine translation and its evaluation: A study. *Artificial Intelligence Review*, *56*(9), 10137-10226. https://doi.org/10.1007/s10462-023-10423-5

Muñoz, J. (2021). *Traducción multilingüe neuronal* [Tesis de maestría]. Universitat Politècnica de València. https://riunet.upv.es/handle/10251/158894

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2024). *A Comprehensive Overview of Large Language Models*. https://doi.org/10.48550/arXiv.2307.06435

Nguyen, T. T. (2019). *Machine translation with transformers* [Tesis de maestría]. University of Stuttgart. http://nbn-resolving.de/urn:nbn:de:bsz:93-opus-ds-106212

Paccotacya Yanque, R. Y. G., & Huanca Anquise, C. A. (2022). *Creación de un corpus de habla del Quechua Collao para el reconocimiento automático de emociones dimen-*

sionales [Tesis de licenciatura]. Universidad Nacional de San Agustín de Arequipa. https://hdl.handle.net/20.500.12773/16383

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A Method for Automatic Evaluation of Machine Translation. En P. Isabelle, E. Charniak & D. Lin (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311-318). Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073135

Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024). *The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities*. https://doi.org/10.48550/arXiv.2408.13296

Popović, M. (2017). chrF++: Words Helping Character n-Grams. En O. Bojar, C. Buck, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn & J. Kreutzer (Eds.), *Proceedings of the Second Conference on Machine Translation* (pp. 612-618). Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-4770

Ranathunga, S., Lee, E.-S. A., Skenduli, M. P., Shekhar, R., Alam, M., & Kaur, R. (2021). *Neural Machine Translation for Low-Resource Languages: A Survey*. https://doi.org/10.48550/arXiv.2106.15115

Raschka, S. (2023, 19 de diciembre). *Practical Tips for Finetuning LLMs Using LoRA (Low-Rank Adaptation)*. Ahead of AI. Consultado el 19 de mayo de 2026, desde https://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-llms

Rivera-Trigueros, I. (2022). Machine translation systems and quality assessment: A systematic review. *Language Resources and Evaluation*, *56*(2), 593-619. https://doi.org/10.1007/s10579-021-09537-5

Runpod. (s.f.). *The AI Developer Cloud*. Runpod. Consultado el 28 de mayo de 2026, desde https://www.runpod.io

Santos, D., Nogueira, V. B., & Quaresma, P. (2025). Parameter Efficient Fine-Tunning of LLMs: Application to Machine Translation from English to Portuguese. *2025 4th International Conference on Computer Technologies (ICCTech)*, 24-28. https://doi.org/10.1109/ICCTech66294.2025.00014

Silva, A., Srivastava, N., Moteu Ngoli, T., Röder, M., Moussallem, D., & Ngonga Ngomo, A.-C. (2024). Benchmarking Low-Resource Machine Translation Systems. En A. K. Ojha, C.-h. Liu, E. Vylomova, F. Pirinen, J. Abbott, J. Washington, N. Oco, V. Malykh, V. Logacheva & X. Zhao (Eds.), *Proceedings of the Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)* (pp. 175-185). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.loresmt-1.18

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, 223-231. https://aclanthology.org/2006.amta-papers.25/

Stahlberg, F. (2020). *Neural Machine Translation: A Review and Survey*. https://doi.org/10.48550/arXiv.1912.02047

Tafa, T. O., Hashim, S. Z. M., Othman, M. S., Alhussian, H., Nasser, M., Abdulkadir, S. J., Huspi, S. H., Adeyemo, S. O., & Bena, Y. A. (2025). Machine Translation Performance for Low-Resource Languages: A Systematic Literature Review. *IEEE Access*, *13*, 72486-72505. https://doi.org/10.1109/ACCESS.2025.3562918

Torero, A. (2002). *Idiomas de los Andes: lingüística e historia*. IFEA, Instituto Francés de Estudios Andinos.

UNESCO. (2025). LT4ALL 2.0: Advancing Humanism through Language Technologies. Consultado el 11 de junio de 2026, desde https://unesdoc.unesco.org/ark:/48223/pf0000392563

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need*. https://doi.org/10.48550/arXiv.1706.03762

Wang, H., Wu, H., He, Z., Huang, L., & Church, K. W. (2022). Progress in Machine Translation. *Engineering*, *18*, 143-153. https://doi.org/10.1016/j.eng.2021.03.023

Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S., & Yang, F. (2025). *Parameter-Efficient Fine-Tuning in Large Models: A Survey of Methodologies*. https://doi.org/10.48550/arXiv.2410.19878

Weaver, W. (1952). Translation. *Proceedings of the Conference on Mechanical Translation*. https://aclanthology.org/1952.earlymt-1.1/

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. https://doi.org/10.48550/arXiv.1609.08144

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J.-R. (2025). *A Survey of Large Language Models*. https://doi.org/10.48550/arXiv.2303.18223

Zhong, T., Yang, Z., Liu, Z., Zhang, R., Liu, Y., Sun, H., Pan, Y., Li, Y., Zhou, Y., Jiang, H., Chen, J., & Liu, T. (2024). *Opportunities and Challenges of Large Language Models for Low-Resource Languages in Humanities Research*. https://doi.org/10.48550/arXiv.2412.04497

Los anexos, panel fotográfico y otros documentos están resguardados en la oficina de repositorio digital institucional en la Biblioteca Central de la Universidad Tecnológica de los Andes