UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERIA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas

Traducción automática neuronal para lengua nativa peruana

Trabajo de Investigación para obtener el Grado Académico de Bachiller en Ingeniería de Sistemas

Autor:

Diego Huarcaya Taquiri

Asesor:

Mg. José Bustamante Romero
Dr. Juan Jesús Soria Quijaite

Lima, octubre 2020

DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

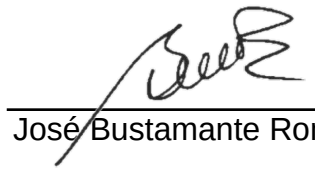
José Bustamante Romero, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que la presente investigación titulada: **"TRADUCCIÓN AUTOMATICA NEURONAL PARA LENGUA NATIVA PERUANA"** constituye la memoria que presenta el estudiante Diego Huarcaya Taquiri para obtener el Grado Académico de Bachiller en Ingeniería de Sistemas, cuyo trabajo de investigación ha sido realizado en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en la ciudad de Lima, a los 14 días del mes de Enero del año 2021.



José Bustamante Romero

ACTA DE SUSTENTACIÓN DE TRABAJO DE INVESTIGACIÓN

En Lima, Ñaña, Villa Unión, a.....los.....09.....día(s) del mes de.....octubre.....del año 2020.... siendo las.....08:30.....horas, se reunieron los miembros del jurado en la Universidad Peruana Unión campus Lima, bajo la dirección del (de la) presidente(a): ..... Dra. Erika Inés Acuña Salinas....., el (la) secretario(a): ..... Mg. Geraldine Verónica Alvizuri Llerena..... y los demás miembros: ..... Mg. Omar Leonel Loaiza Jara.....y el (la) asesor(a): Mg. José Bustamante Romero y co asesor(a): ....Dr. Juan Jesús Soria Quijaite .... con el propósito de administrar el acto académico de sustentación del trabajo de investigación titulado: "Traducción Automática Neuronal para Lengua Nativa Peruana".

.....de los (las) egresados (as): a)..... Diego Huarcaya Taquiri.....

.....b).....

..... conducente a la obtención del grado académico de Bachiller en

.....Ingeniería de Sistemas.....
(Denominación del Grado Académico de Bachiller)

El Presidente inició el acto académico de sustentación invitando ...al... candidato(a)/s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por ...el... candidato(a)/s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato/a (a): ..... Diego Huarcaya Taquiri .....

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	18	A-	Con nominación muy bueno	Sobresaliente

Candidato/a (b): .....

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó ...al... candidato(a)/s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente
Dra. Erika Inés Acuña Salinas

Secretario
Mg. Geraldine Verónica Alvizuri Llerena

Asesor
M.Sc. Fredy Abel Huanca Torres

Miembro

Miembro
Mg. Cynthia Carol Acuña Salinas

Candidato/a (a)
Jos Lee Josue Villegas Pacheco

Candidato/a (b)

Contenido

Traducción Automática Neuronal para Lengua Nativa Peruana

Diego Huarcaya[1][0000-1111-2222-3333]

[1] Universidad Peruana Unión, Carretera Central Km 19.5 Ñaña, Lima, Perú
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
diegohuarcaya@upeu.edu.pe

Abstract. La traducción automática (MT) es un subcampo de lingüística computacional que se enfoca en traducir un texto de un idioma a otro. Con el creciente auge del aprendizaje profundo, la Traducción Automática Neuronal (NMT) ha conducido a mejoras notables frente a las técnicas convencionales de Traducción Automática Estadística (SMT) y basadas en reglas. En este presente artículo se aplicó técnicas de Traducción Automática Neuronal para crear un modelo de Traducción Automática Neuronal para la traducción entre el idioma español a quechua Chanka. Con respecto a la arquitectura del modelo, se usó la arquitectura basados en mecanismos de atención denominada Transformers. Además, este trabajo proporciona nuevos recursos que comprenden un nuevo corpus con 119, 000 traducciones paralelas. En cuanto a los resultados experimentales indican en términos de Bi-Lingual Evaluation Understudy (BLEU) un puntaje de 39,5.

Keywords: Traducción Automática, Redes neuronales, Lengua nativa peruana.

Neural Machine Translation for Peruvian Native Language

Diego Huarcaya[1][0000-1111-2222-3333]

[1] Universidad Peruana Unión, Carretera Central Km 19.5 Ñaña, Lima, Perú
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
diegohuarcaya@upeu.edu.pe

Abstract. Machine translation (MT) is a subfield of computational linguistics that focuses on translating a text from one language to another. With the growing rise of deep learning, Neural Machine Translation (NMT) has led to notable improvements over conventional Statistical Machine Translation (SMT) and rule-based techniques. In this present article, Neural Machine Translation techniques were applied to create a Neural Machine Translation model for translation from Spanish to Quechua Chanka. Regarding the architecture of the model, the architecture based on attention mechanisms called Transformers was used. Furthermore, this work provides new resources comprising a new corpus with 119,000 parallel translations. Regarding the experimental results, they indicate in terms of Bi-Lingual Evaluation Understudy (BLEU) score 39,5.

Keywords: Machine Translation, Neural Networks, Native Peruvian Language.

1 Introducción

En el Perú, según cifras oficiales, se tienen aproximadamente 47 lenguas indígenas u originarias y todas ellas son importantes por ser vehículo de comunicación de todas las culturas. A través del uso de sus lenguas, los pueblos conservan y transmiten sus afectos, sus tradiciones, su cosmovisión, sus propios valores y sus conocimientos a las siguientes generaciones y al mundo. [1]

La Educación Intercultural Bilingüe (EIB), como política educativa peruana, es para todos y promueve aprendizajes de la lengua originaria y del castellano desde el nivel inicial. En contextos indígenas, hay varias evidencias y testimonios de que se obtienen mejores resultados cuando los profesores saben y usan la lengua de la comunidad en la escuela y promueven aprendizajes desde su cultura hacia los demás. [1]

La aplicación apropiada de las tecnologías del lenguaje humano, incluyendo las herramientas de traducción automática, brinda una infinidad de oportunidades para la revitalización, la difusión y el aprendizaje de nuestros idiomas, para la comunicación intergeneracional entre abuelos monolingües en una lengua indígena y sus nietos monolingües que hablan español, así como para el acceso a servicios y la democratización de la información [2].

La lingüística computacional de lenguas de bajos recursos, requiere de técnicas más allá del entrenamiento básico de modelos [2]. Procesar una nueva lengua conlleva a nuevos desafíos (sistemas fonológicos especiales, problemas de segmentación, estructuras gramaticales, lenguaje no escrito entre otros), Por otra parte, la falta de recursos requiere, por su parte, innovación en las metodologías de recolección de datos o modelos para los cuales la información es compartida entre varias lenguas.

Esta investigación tiene como objetivo la construcción de una versión inicial de un modelo de Traducción Automática Neuronal para una lengua de bajos recursos partiendo desde la recolección de datos hasta la evaluación del modelo. El caso de estudio tiene un lenguaje de origen como de objetivo, siendo este el español y el quechua Chanka respectivamente.

Este artículo está organizado de la siguiente manera. En la sección 2 se muestra lo más relevante en cuanto tipos de traducción automática. La sección 3 hace una descripción de las arquitecturas y definición de Traducción Automática Neuronal. En la sección 4 describe teóricamente los materiales y métodos empleados. La sección 5 muestra los resultados obtenidos tras la aplicación de los métodos y materiales mencionados en la anterior sección. Finalmente, la sección 6 muestra las conclusiones alcanzadas.

2 Antecedentes

2.1 Traducción Automática basada en reglas (RBMT)

Consiste en la recopilación de reglas gramaticales, léxico para luego ser procesadas por programas de software. Es fácil de escalar y también de modificar. La base del conocimiento presente en este tipo de traducción esta dado generalmente por lingüistas. Las reglas desempeñan un papel muy importante en las diversas etapas de la traducción tanto como en el: procesamiento sintáctico, interpretación semántica y procesamiento contextual del lenguaje. [3]

La construcción de este tipo de traductores automáticos demanda una gran cantidad de tiempo y bastos recursos lingüísticos, como resultado es muy caro. Para poder mejorar la calidad de traducción es necesario modificar las reglas y esta requiere más conocimiento lingüístico, además de no tener la garantía de que modificar una o más reglas pueda mejorar la calidad de la traducción. [4]

2.2 Traducción Automática Estadistica (SMT)

La Traducción Automática Estadística (SMT), es un paradigma de Traducción Automática que se basa en modelo estadísticos aprendidos de corpus de texto paralelo y algoritmos de decodificación para traducir automáticamente de un lenguaje natural a otro. [5] Es de aquí que aparecen los primeros modelos de traducción junto con el modelado de lenguaje. El modelo de lenguaje nos proporciona probabilidades para una cadena de palabras u oraciones que se puede denotar como por $\Pr(S)$ donde este último es la probabilidad de que una cadena de palabras de origen S ocurra. El modelo de traducción estadística incluye estas probabilidades $\Pr(T|S)$ donde la probabilidad condicional de una oración objetivo T ocurrirá en un texto de destino que traduce un texto que contiene una oración fuente S, a la actualidad estos tipos de Traductores automáticos fueron reformulados usando la regla de Bayes. [4]

Por otra parte, este tipo de traducciones requieren grandes recursos computacionales, carece de fundamento lingüístico ya que está basado en probabilidades y en lenguajes de bajos recursos ocurre problemas con la morfología entre las lenguas a traducir. [4]

3 Traducción Automática Neuronal (NMT)

La Traducción Automática de texto o voz de un idioma a otro es una de las metas más desafiantes para las maquinas. [6] Este es un método basado en Redes Neuronales, propuesto por (Kalchbrenner and Blunsom, 2013), (Cho et al., 2014) y (Sutskever et al., 2014) y presentado por Google en 2016. Es el arte de usar modelos de Redes Neuronales Artificiales (ANN) para aprender modelos estadísticos de traducción automática. [6] Las Redes Neuronales analizan una gran cantidad de datos del corpus para luego aprender las reglas gramaticales automáticamente a partir de los ejemplos presentes en el corpus.

La arquitectura de los modelos de NMT generalmente consisten en un codificador y decodificador, estas presentan Redes Neuronales Recurrentes (RNN), estos últimos permiten predecir la palabra que aparecerá en la siguiente secuencia a través de la asociación de la palabra previa. [18] Con respecto a la tarea de traducción la RNN toma el idioma de origen como la secuencia de entrada y el idioma objetivo como la secuencia de salida dando como resultado una traducción holística en lugar de una traducción palabra por palabra.

4 Materiales y Métodos

En esta sección se recopilan y se exponen, los materiales y métodos aplicados en el desarrollo de esta investigación. En el apartado de métodos se especifican detalladamente el procedimiento seguido para alcanzar el objetivo propuesto.

4.1 Materiales

Python 3. Se optó por este lenguaje de programación debido a su gran cantidad de librerías disponibles y la potencia del lenguaje. [8] En comparación con otros lenguajes de programación populares para Inteligencia Artificial como C++ o Java, Python es ampliamente usado para computación científica, computación avanzada como aprendizaje de máquina, estos recursos de código abierto presentan bibliotecas que suministran herramientas analíticas, algorítmicas, matriciales que nos permiten alcanzar el objetivo propuesto en esta investigación.

Google Colaboratory. Conocido también como Colab, es un servicio en la nube basado en Jupyter Notebooks con el objetivo de difundir la educación y la investigación en el campo de la Inteligencia Artificial. [9] Nos proporciona entornos y tiempos de ejecución para Python en sus versiones 2 y 3, con las bibliotecas esenciales para Aprendizaje Automático e Inteligencia Artificial como Tensorflow, Keras, Scikit Learn. Este servicio provee tiempos de ejecución con CPU. GPU y TPU.

Tensorflow. Permite a desarrolladores e investigadores experimentar con algoritmos de entrenamiento, optimización e inferencia [10]. La importancia de Tensorflow en esta investigación radica en su API de alto nivel denominado Keras este último nos permite la creación rápida de prototipos, la muestra de información clara y procesable sobre el manejo de errores. Además, incluye métricas, funciones de perdida y modelos pre configurados que en adelante nos ayudara a validar el modelo de NMT.

4.2 Métodos

Esta investigación consta de 5 fases: La primera fase explica el proceso y las técnicas usadas para la recolección de los datos; la etapa de pre procesamiento expone las herramientas, técnicas y recursos aplicados al pre procesamiento de estos datos; en la tercera fase se vectoriza los tokens con el objetivo de poder representar las palabras y sus relaciones aplicando técnicas matemáticas para ser entrenadas en la siguiente etapa y finalmente los resultados son evaluados usando la métrica BLEU en la etapa de evaluación. Los métodos desarrollados están resumidos en la Fig 1.

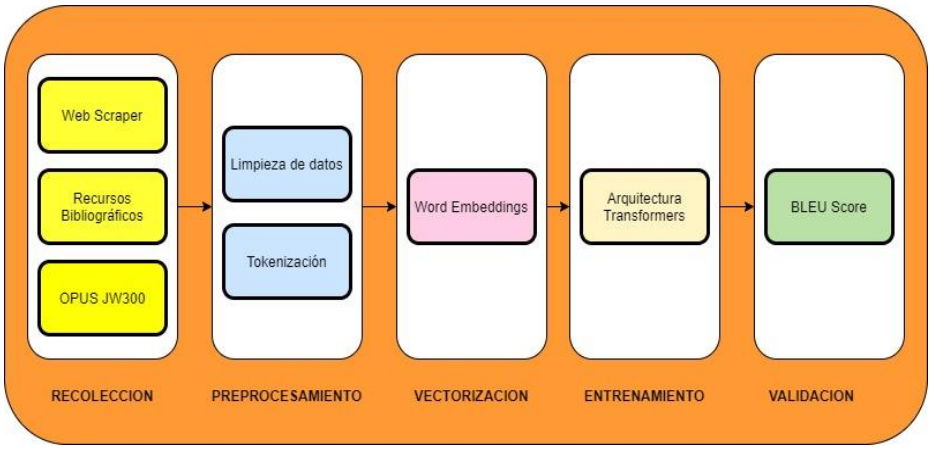


Fig. 1. Flujo de trabajo y el diseño de estudio.

Recolección de datos

Un componente esencial de un sistema de Traducción automática es el corpus paralelo [11]. Este conjunto de datos es indispensable para el proceso de Traducción Automática, ya que permite la extracción de características y comportamiento de las unidades léxicas de los lenguajes a procesar [12] Dada la naturaleza lingüística de esta investigación el tipo de dato a recolectar serán oraciones y palabras con su correspondiente traducción paralela en adelante al conjunto de estos se denominará corpus paralelo. Los textos se extraerán básicamente de diversas fuentes electrónicas y bibliográficas debido a que se trata de una lengua de bajos recursos. A continuación, se explica las diferentes técnicas y recursos a la que se recurrió para cumplir con este propósito.

Web Scrapping. También conocida como extracción web o harvesting, es una técnica de extracción de datos desde la Web para luego almacenar estos en una base de datos o en algún archivo para su posterior procesamiento o análisis. [13] En la investigación, esta técnica tendrá la finalidad de automatizar la extracción de traducciones paralelas presentes en la página web runasimi.org para luego almacenarlas en un archivo de texto plano.

Fuentes Bibliograficas. En este caso los textos serán extraídos de manuales pertenecientes al Gobierno Peruano. Los textos pertenecen específicamente de los Ministerios de Justicia y Educación, estos textos están disponibles en línea, para estas fuentes se extraerá las traducciones manualmente. Cabe destacar que no se recurrió al uso del OCR (Optical Character Recognition) debido a la distorsión de textos que este presentaba tras extraer las traducciones paralelas.

Opus JW300. Es una colección masiva de textos paralelos para más de 300 lenguajes distintos cuyo principal objetivo es facilitar el Procesamiento de Lenguaje Natural (NLP) plurilingüe. [14] JW300 comprende un total de 1,335,376 artículos con un poco

más de 109 millones de oraciones, y 1.48 billones de tokens. El conjunto de datos es completamente recolectado de todas las publicaciones de la página web jw.org, la mayoría de los textos provienen de revistas como Awake y Watchtower cuyo contenido de estos últimos es esencialmente religioso. Este recurso nos permitirá poder incrementar la cantidad de traducciones paralelas contenidas en el corpus.

Preprocesamiento de datos

La calidad del conocimiento extraído depende en gran medida de la calidad de los datos. Generalmente, estos datos se ven afectados por valores negativos como ruido, valores perdidos, datos superfluos, o una longitud de caracteres demasiado amplio, lo que conduce en buena parte de casos a una baja calidad del modelo. [13] Para pre procesar el corpus paralelo se realizarán principalmente dos principales tareas que se presentan a continuación:

Limpieza.

Es un proceso utilizado para determinar datos inexactos, incompletos o irrazonables para luego mejorar la calidad de este, mediante la corrección de errores u omisiones que previamente fueron detectados. [15] Para esta tarea se harán uso de correctores ortográficos para validar las oraciones en la lengua española mientras para la lengua quechua se hará de forma manual, para remover los caracteres especiales como viñetas, corchetes, llaves y sangrías se recurrirán al uso de expresiones regulares, cabe destacar que esta última también nos permitirá crear un espacio entre la palabra y el signo de puntuación continuo con la finalidad de facilitar el proceso de tokenización del corpus.

Tokenización.

Es una técnica en Procesamiento de Lenguaje Natural (NLP) que divide el documento en tokens que pueden ser palabras o frases. [16] La utilidad de esta técnica radica en la identificación de palabras significativas en una oración. Los tokens están separados por espacios en blanco, saltos de línea o signos de puntuación. Los espacios en blanco y las puntaciones no se incluyen en los tokens resultados. En la investigación se convertirá los textos en secuencias de palabras separadas por espacios. Estas secuencias se dividen en lista de tokens cada token con su correspondiente indexación, estos tokens serán codificados en formato UTF-8, el motivo de usar este formato para la codificar los tokens es debido a la presencia tanto de la consonante “Ñ” y las tildes en el corpus paralelo. Para finalizar se hace la división del corpus usando el principio de Pareto siendo este 80% de los datos para el entrenamiento, 10% para la validación y 10% para el test que se usara como referencia en la métrica BLEU.

Vectorización

La representación de palabras es un componente critico en mayoría de sistemas de Procesamiento de Lenguaje Natural (NLP), De manera común se suele representar palabras como índices en un vocabulario, pero esta no logra capturar la riqueza léxica y estructural. [23] En este sentido los modelos basados en vectores hacen mucho mejor

esta tarea. Codifican similitudes continuas entre palabras usando distancia o ángulos entre las palabras vectorizadas en un espacio dimensional.

Word Embeddings

Es una técnica de aprendizaje de características cuyo objetivo es mapear palabras de un vocabulario en vectores de números reales en un espacio dimensional. [22] Tales representaciones de espacio continuo se pueden calcular para capturar información tanto sintáctica como semántica a partir de las palabras. Debido a que las palabras presentes en el corpus paralelo ocurren en contextos similares las palabras resultantes son semánticamente cercanas después del entrenamiento. En esta investigación se hace uso de técnicas matemáticas como el análisis de componentes principales (PCA) y la distancia Euclidiana [24] para poder representar las palabras y sus relaciones de las palabras contenidas en el corpus.

Entrenamiento del modelo NMT

Transformers.

Propuesta por Vaswani et al. (2017), es un modelo de Red Neuronal Recurrente usado para ser la más popular y más robusta arquitectura para la estructura del codificador – decodificador para la solución de problemas relacionados con la Traducción Automática Neuronal. [6] Este modelo emplea mecanismos de Auto-Atención que permite tanto al codificador como el decodificador tener en cuenta cada palabra de toda la secuencia de entrada. Como también ha demostrado ser muy prometedor para lenguajes de bajos recursos [17] Se optó por el uso de esta arquitectura debido a que el quechua es una lengua de bajos recursos y por sus capas de Multi Atención que permiten analizar mejor el contexto de la oración y predecirla según este último, cabe destacar que para obtener la cantidad de épocas adecuadas se usaron callbacks propias de Tensorflow. Para finalizar se descartó el uso de arquitecturas para modelos preentrenados como BERT (Bidirectional Encoder Representations from Transformers) debido a que fue entrenado principalmente con lenguajes de altos recursos y para ciertas tareas específicas.

Validación del modelo

BLEU (Bilingual Evaluation Understudy).

Métrica que será usada para evaluar la calidad de texto traducido y ampliamente usada en la comunidad de Traducción Automática. BLEU es usada para calcular la precisión de las oraciones traducidas comparándolas con las traducciones de referencia hecha por humanos [18]. Esta métrica funciona calculando la media geométrica de la precisión p_n de n-gramas, donde normalmente van entre $1 \leq n \leq 4$, comparando la traducción propuesta con una o más traducciones de referencia. En la investigación esta métrica nos permitirá evaluar la calidad de traducción teniendo como n gramas las traducciones entrenadas y como referencia las traducciones de validación definidas en la división de datos. La escala de esta métrica va desde 0 a 100.

$$\text{BLEU} = \min\left(\exp\left(1 - \frac{\text{r}}{\text{c}}\right), 1\right). \exp(\sum_{n=1}^{N} \frac{logp_n}{N}) (1)$$

Character n –gram F-Score (chrF).

Se optó por el uso de esta metrica para la evaluación del modelo por las siguientes razones; es independiente del lenguaje, independiente de la tokenizacion y muestra una aceptable correlación entre los juicios humanos y obtenidos además de ser aplicados en diferentes sistemas de escritura como el árabe y el chino. [25].

5 Resultados

5.1 Recolección de datos

Primero se construyó un Web Scrapper para la página donde se extrajo los datos con esta herramienta automatizada. Por otra parte, se obtuvo de manera manual de recursos como Manuales de Administración de Justicia del estado peruano [19], Manuales de Quechua Chanka del Ministerio de Educación [20]. Adicionalmente se usó la herramienta de corpus paralelo para lenguajes de bajos recursos OPUS JW300 que viene a ser una colección de corpus paralelos abiertamente disponibles y que está ampliamente usado en trabajos en traducción automática e investigación translingüística. Después de este procedimiento se obtuvieron 145, 245 traducciones paralelas.

Web Scrapping	Recursos Bibliográficos	Opus JW300	Total
6565	7043	124 437	145 245

Tabla 1. Conteo de traducciones paralelas obtenidas por cada recurso

5.2 Preprocesamiento de datos

Para la construcción del corpus final, se eliminaron los valores vacíos, así como las traducciones paralelas duplicadas para luego hacer uso de expresiones regulares (RegEx), como también se hizo una verificación manual de las traducciones obteniendo un total de 122, 060 frases paralelas. Los datos se dividieron para entrenamiento que fueron 102 060 oraciones, 10.000 para la validación y otros 10.000 para las pruebas. El corpus fue codificado en formato UTF – 8.

Corpus	Oraciones Paralelas
No filtrado	145 245
Traducciones en blanco	20
Traducciones incorrectas	23 185
Filtrado	122 060

Entrenamiento	102 060
Validación	10 000
Test	10 000

Tabla 2. Estadísticas del corpus paralelo antes y después del filtrado.

Tokenizacion.

Tras el proceso de limpieza de los datos se pasó a tokenizar. Para esta tarea se hizo uso de la función Tokenizer de Tensorflow, tras aplicar esta función se indexó las palabras y se tokenizó las oraciones.

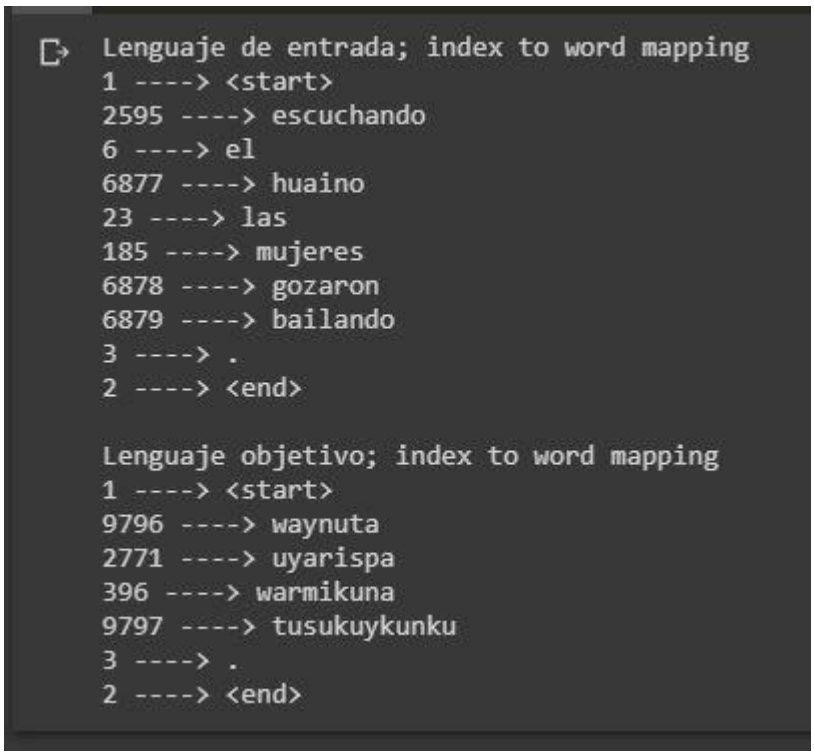


Fig. 2. Ejemplo de oración tokenizada e indexada, cada token se le asigna una posición.

5.3 Vectorización

Para capturar las similitudes y características semánticas se segmentó las oraciones en tokens, tras esto se pasó a vectorizar los tokens, a partir de este último se produjeron las representaciones de los Word Embeddings en un espacio tridimensional haciendo uso de técnicas matemáticas como Análisis de componentes principales (PCA) y la

distancia Euclidiana, para visualizar los Embeddings se hizo uso de Tensorflow Projector.

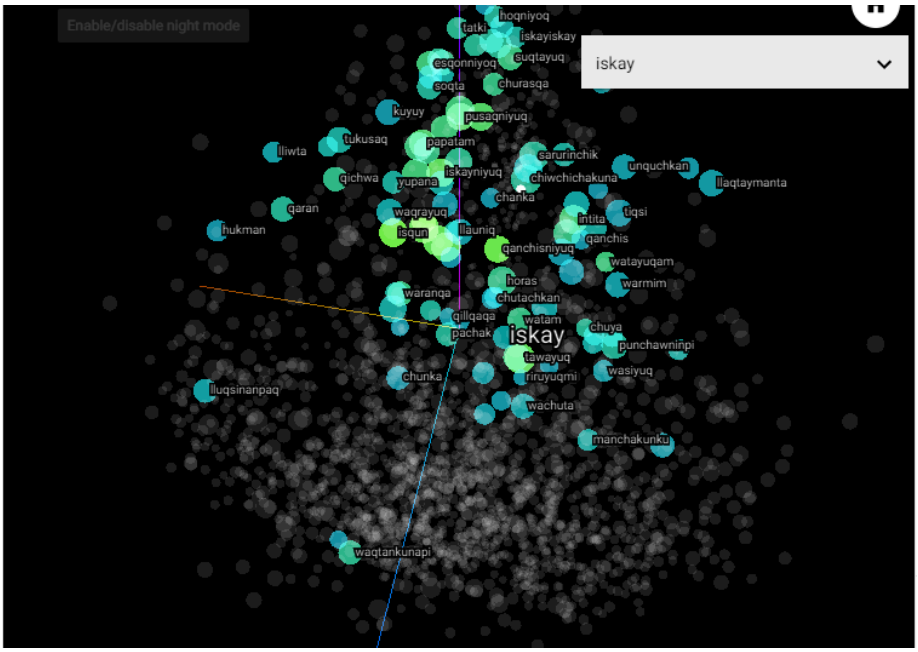


Fig. 3. Visualizador de Word Embeddings en un espacio tridimensional muestra la palabra "Iskay" y las palabras conceptualmente cercanas a esta el verde claro representa ser cercana mientras la celeste muestra estar débilmente relacionada a esta palabra.

5.4 Entrenamiento del modelo

Para el desarrollo del Modelo se usó Keras con Tensorflow como backend para la arquitectura Transformers. Esta arquitectura está implementada está basada en módulos de codificación, decodificación y una función de activación Softmax.

Parámetros	Valor
Capas multihead	8
Embeddings	512
Batch de Tokens	4096 (1 GPU)
Numero de Epocas	25
Tokenizador	BPE (Byte Pair Encoding)
Optimizador	Adam

Tabla 3. Configuración de los parámetros usados en la red neuronal

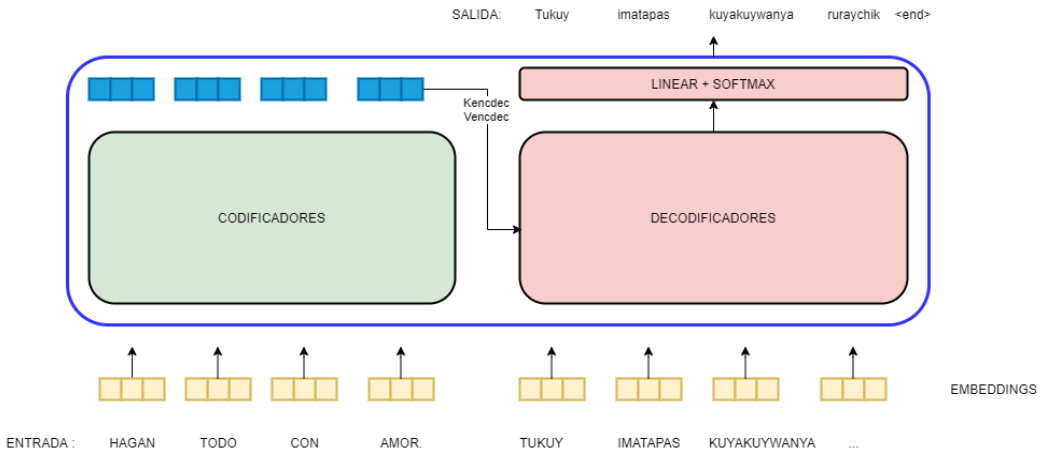


Fig. 4. Funcionamiento del modelo de traducción bajo la arquitectura Transformers, el color azul representa los vectores de Atención.

La siguiente muestra la evolución de la métrica de validez en el corpus paralelo durante 13 mil iteraciones.

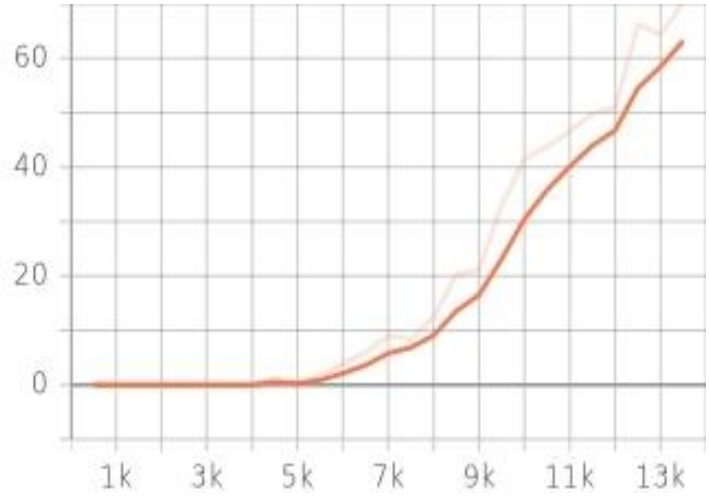


Fig. 5. Evolución de la validez durante el entrenamiento.

Por contraparte las imágenes mostradas a continuación muestran la evolución con respecto a la función de error para 13 mil iteraciones, el primero muestra el error en agrupaciones de datos mientras el otro por cada dato.

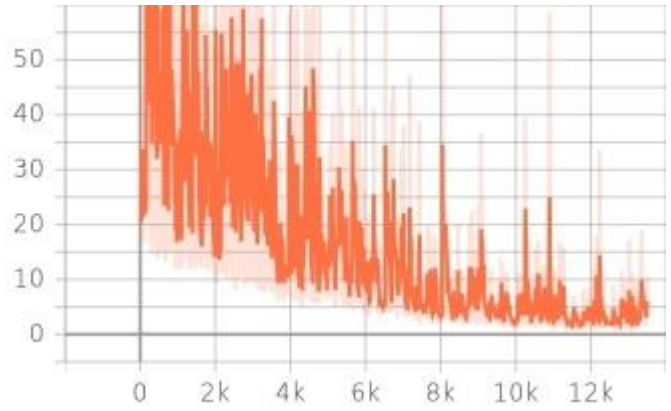


Fig. 6. Evolución de la función de error por lotes (batch).

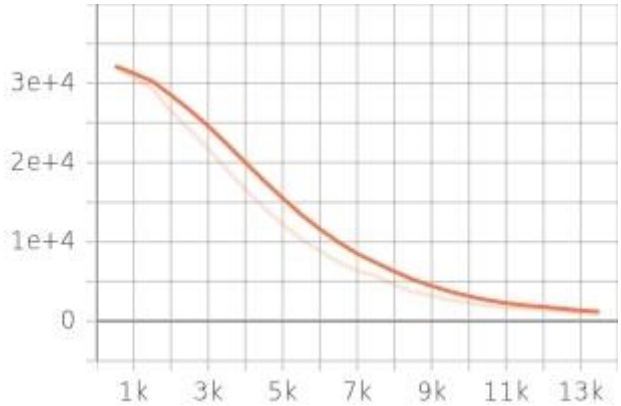


Fig. 7. Evolución de la función de error durante el entrenamiento

5.5 Validación del modelo

Para la validación del modelo se usó la métrica BLEU, se hicieron los test con el corpus en sus distintas cantidades, se pudo apreciar mientras la cantidad de traducciones paralelas sea más grande y la calidad de estas sea correcta la puntuación aumenta. Se usó solamente esta métrica debido a que las demás métricas fueron y están hechas para lenguajes de altos recursos además de las diferencias léxicas que estas presentan.

BLEU:

BLEU Score Test 1	BLEU Score Test 2	BLEU Score Test 3	BLEU Score 4
8.43	15.2	20.1	39.50174

Table 4. Puntuación BLEU obtenidas.

Test 1: Pruebas con el corpus inicial
Test 2: Pruebas con mayor cantidad de traducciones paralelas y épocas de entrenamiento mayores.
Test 3: Pruebas con el corpus final.
Test 4: Pruebas con el corpus y modificando los hiperparametros.

ChrF:

Épocas	Puntuación ChrF
5	0.0
10	0.8
20	0.19
25	0.24

Table 5. Puntuación ChrF por distintas épocas de entrenamiento.

6 Conclusiones

Este artículo reúne el desafío de abordar la traducción del español a quechua Chanka con las últimas técnicas de Traducción de Máquina. Para el desarrollo del modelo, se creó un nuevo corpus usando técnicas de extracción de datos y por otra parte se usó el corpus ya existente proporcionado por OPUS JW300. En este trabajo el modelo fue creado con la arquitectura Transformers. Entre las principales limitaciones encontradas en este estudio fueron el tamaño del corpus paralelo disponible y la gran diferencias léxicas y gramaticales entre ambos idiomas. Los resultados muestran una aproximación y se alcanzó la puntuación de 39.5 en la escala BLEU como 0.24 en la métrica ChrF para la tarea de español a quechua Chanka.

Referencias

1. Ministerio de Cultura: 10 cosas que debes saber sobre las Lenguas Indígenas peruanas y sus hablantes. Minist. Cult. 12 (2013).
2. Camacho, L., Zevallos, R.: Siminchikkunarayku: Lingüística computacional para la revitalización y el poliglotismo. (2019).
3. Arnold, D., Sadler, L.: Machine Translation: an Introductory Guide. (2017).
4. Costa-Jussà, M.R. et al.: Study and comparison of rule-based and statistical catalan-spanish machine translation systems. Comput. Informatics. 31, 2, 245–270 (2012).
5. Xiong, D., Zhang, M.: Linguistically motivated statistical machine translation: Models and algorithms. (2015). https://doi.org/10.1007/978-981-287-356-9.
6. Nguyen, T.T., Thesis, M.: Machine Translation with Transformers. (2019).

7. Vaswani, A. et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. 2017-Decem, Nips, 5999–6009 (2017).
8. Duque, R. G. (2011). Python para todos
9. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283).
10. Carneiro, T. et al.: Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. IEEE Access. 6, 61677–61685 (2018). https://doi.org/10.1109/ACCESS.2018.2874767.
11. Mager, M. et al.: Challenges of language technologies for the indigenous languages of the Americas. 55–69 (2018).
12. Nguyen, V. et al.: Towards State-of-the-art English-Vietnamese Neural Machine. 120–126 (2015).
13. Zhao, B.: Encyclopedia of Big Data. Encycl. Big Data. May 2017, (2020). https://doi.org/10.1007/978-3-319-32001-4.
14. Agic, Ž., Vulic, I.: JW300: A wide-coverage parallel corpus for low-resource languages. ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. 3204–3210 (2020).
15. Arthur D, C.: Principles and Methods of Data Cleaning. Glob. Biodivers. Inf. Facil. 70, 9, 48–53 (2005). https://doi.org/10.1126/science.328.5974.18-c.
16. Joseph, J., Jeba, J.R.: Information Extraction using Tokenization and Clustering Methods. Int. J. Recent Technol. Eng. 8, 4, 3690–3692 (2019).
17. van Biljon, E. et al.: On Optimal Transformer Depth for Low-Resource Language Translation. 1–6 (2020).
18. Smith, A., Hardmeier, C.: BLEU Is Not the Colour : How Optimising BLEU Reduces Translation Quality. 2013–2015 (2003).
19. El, M.P. et al.: Manual para el empleo del Quechua Chanka en la administración de justicia. (2014).
20. Vizcardo Rozas, N.: Manual Auto Instructivo Curso “Quechua Básico.” 1–66 (2016).
21. Herrera, F.: Big Data: Preprocesamiento y calidad de datos. Novática. 237, 17 (2016).
22. Znotiņš, A.: Word embeddings for Latvian natural language processing tools. Front. Artif. Intell. Appl. 289, 167–173 (2016). https://doi.org/10.3233/978-1-61499-701-6-167.
23. Maas, A.L. et al.: Learning word vectors for sentiment analysis. ACL-HLT 2011 - Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol. 1, 142–150 (2011).
24. Korenius, T. et al.: On principal component analysis, cosine and Euclidean measures in information retrieval. Inf. Sci. (Ny). 177, 22, 4893–4905 (2007). https://doi.org/10.1016/j.ins.2007.05.027.
25. Maja, P.: CHRF: character n-gram F-score for automatic MT evaluation. 31, 4, 344–345 (2015). https://doi.org/10.1080/1472586x.2015.1113070.