UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

TESIS DE GRADO

"TRADUCTOR AUTOMÁTICO ESPAÑOL – QUECHUA, BASADO EN EL PROCESAMIENTO DEL LENGUAJE NATURAL"

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POR: RENÉ ALEX APANQUI OTOYA
TUTOR: M. Sc. ROSA FLORES MORALES

LA PAZ - BOLIVIA
2021

HOJA DE CALIFICACIONES

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMÁTICA

Tesis de grado:

TRADUCTOR AUTOMÁTICO ESPAÑOL – QUECHUA, BASADO EN EL PROCESAMIENTO DEL LENGUAJE NATURAL

Presentado por: René Alex Apanqui Otoya

Para optar el grado Académico de Licenciado en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral:

Nota Literal:

Ha sido:

Director de la carrera de Informática: Ph.D. Jose Maria Tapia Baltazar

Tutor: M.Sc. Rosa Flores Morales

Tribunal:

Tribunal:

Tribunal:

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMÁTICA

LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.

b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.

c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

Dedicatoria

A mis padres Alejandro Apanqui Quispe y Sabina Otoya Mollo por darme la vida y enseñarme a sobrellevar las adversidades de la vida, también a mis hermanos y hermanas por su constante apoyo.

También dedicar este trabajo a todos mis compañeros que conocí en la universidad que con el trascurso del tiempo se fueron convirtiendo en mis mejores amigos y que con ellos formamos una segunda familia.

Agradecimientos

Primeramente agradecer a Dios por ayudarme en todo momento y levantarme en los momentos difíciles de mi vida y darme la oportunidad de conservar a mi familia en toda esta trayectoria.

También dar un especial agradecimiento a mi familia, por todo el amor que me brindan y toda su colaboración en todas las etapas de mi vida, tanto escolar, como universitaria, por sus palabras de aliento y sobre todo sus consejos.

De la misma manera un agradecimiento sincero a mi tutor metodológico M.Sc. Rosa Flores Morales por brindarme todo su apoyo, sus conocimientos, su experiencia, sus consejos, paciencia, comprensión y su confianza hacia mi persona, su persona y carisma estará siempre conmigo.

Gracias.

lexingthon77@gmail.com

RESUMEN

En este trabajo, se presenta un traductor automático entre los idiomas español y quechua por medio del Procesamiento de Lenguaje Natural. Para su desarrollo, se decidió utilizar el modelo de Traducción Automática basada en redes Neuronales NMT, implementando dos corpus paralelos de cada idioma con 1,919 frases alineadas en el algoritmo *Transformer*, para su respectivo proceso de entrenamiento y aprendizaje de las redes neuronales, usando las herramientas básicas para el procesamiento de lenguaje natural, como es un tokenizador, codificador, normalizador y decodificador. Estas herramientas las implementamos guiándonos de la metodología de Proceso Cíclico expuesta en el marco teórico. Los resultados obtenidos no son del todo óptimos pero son aceptables tomando en cuenta el escaso corpus bilingüe con el que se cuenta. En conclusión, el uso de una lengua aglutinada como es el quechua en un sistema NMT, debe ser más estudiada para generar mejores resultados, pero abre la puerta para que nuevos idiomas puedan ser traducidos, sobre todo si son semejantes como el quechua, aimara o el guaraní.

Palabras clave: Procesamiento de lenguaje natural, quechua, redes neuronales, corpus paralelo, traducción automática.

ABSTRACT

In this work, an automatic translator between the Spanish and Quechua languages is presented through Natural Language Processing. For its development, it was decided to use the Machine Translation model based on NMT Neural networks, implementing two parallel corpus of each language with 1,919 sentences aligned in the Transformer algorithm, for their respective training and learning process of neural networks, using the tools basic to natural language processing, such as a tokenizer, encoder, normalizer, and decoder. We implement these tools guided by the Cyclical Process methodology exposed in the theoretical framework. The results obtained are not entirely optimal but they are acceptable taking into account the limited bilingual corpus that is available. In conclusion, the use of an agglutinated language such as Quechua in an NMT system should be further studied to generate better results, but it opens the door for new languages to be translated, especially if they are similar such as Quechua, Aymara or the Guarani.

Keywords: Natural language processing, quechua, neural networks, parallel corpus, machine translation.

Índice de Contenido

CAPÍTULO I. MARCO REFERENCIAL

1.1. INTRODUCCIÓN **2**

1.2. ANTECEDENTES **3**

1.3. PLANTEAMIENTO DEL PROBLEMA **4**

1.4. OBJETIVOS **6**

1.4.1. Objetivo general **6**

1.4.1. Objetivos específicos **6**

1.5. JUSTIFICACIÓN **6**

1.6. LÍMITES Y ALCANCES **7**

1.6.1. Límites **7**

1.6.2. Alcances **7**

1.7. DISEÑO METODOLÓGICO **8**

CAPÍTULO II. IDIOMA QUECHUA

2.1. ANTECEDENTES **10**

2.2. DESCRIPCIÓN LINGÜÍSTICA **11**

2.2.1. Fonología **11**

2.2.2. Morfología **13**

2.2.3. Escritura normalizada **14**

CAPÍTULO III. ESTADO DEL ARTE

3.1. AVANCES TECNOLÓGICOS DE LA TRADUCCIÓN AUTOMÁTICA **18**

3.2. ANTECEDENTES DE TRADUCTORES AUTOMÁTICOS DEL ESPAÑOL AL QUECHUA **21**

3.2.1. Traductor automático en línea del español al quechua (Apertium) **22**

3.2.2. Traductor morfológico del castellano al quechua (TECSUP)..... 23

3.3. MODELOS DE TRADUCCIÓN AUTOMÁTICA..... 24

3.4. TRADUCCIÓN AUTOMÁTICA BASADA EN REDES NEURONALES 28

3.4.1. Modelo Transformer 30

3.4.1.1. Arquitectura del modelo Transformer 30

CAPÍTULO IV. MARCO TEÓRICO

4.1. PROCESAMIENTO DEL LENGUAJE NATURAL (NLP)..... 36

4.1.1. Orígenes de NLP 36

4.2. REDES NEURONALES CONVOLUCIONALES (CNN) 37

4.2.1. Redes Neuronales Convolucionales para NLP 39

4.3. LENGUAJE DE PROGRAMACIÓN PYTHON..... 41

4.4. TENSORFLOW..... 41

4.5. KERAS..... 41

4.6. METODOLOGÍA DE DESARROLLO..... 42

4.6.1. Proceso Cíclico..... 43

4.7. CRITERIOS DE EVALUACIÓN DE CALIDAD DEL TRADUCTOR AUTOMÁTICO 46

4.7.1. Evaluación humana 47

4.7.2. Métodos automáticos 48

CAPÍTULO V. MARCO APLICATIVO

5.1. ANÁLISIS PARA EL DESARROLLO DEL PROTOTIPO 51

5.2. DISEÑO Y MODELACIÓN DEL TRADUCTOR AUTOMÁTICO..... 52

5.2.1. Iteración 1 52

5.2.1.1. Compilación de datos 52

5.2.1.2. Proceso de entrenamiento del motor de traductor automático 56

5.2.1.3. Evaluación 58

5.2.2. Iteración 2 59

5.2.2.1. Pre edición del texto original 60

5.2.2.2. Traducción 60

CAPÍTULO VI. PRUEBAS Y RESULTADOS

6.1. EVALUACIÓN 64

6.1.1. Evaluación humana 64

6.2. PRUEBAS DEL PROTOTIPO 65

6.2.1. Estrategias de prueba 66

6.3. COMPARACIÓN DE RESULTADOS 69

CAPÍTULO VII. CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES 72

7.2. RECOMENDACIONES 73

BIBLIOGRAFÍA 75

Índice de Figuras

CAPÍTULO III

Figura 3.1 Módulos del sistema de traducción Automática de Apertium **22**

Figura 3.2 Proceso de traducción del traductor morfológico TECSUP **23**

Figura 3.3 Triángulo de Vauquois **25**

Figura 3.4 Ejemplo de oraciones en un corpus paralelo del inglés al español **26**

Figura 3.5 Ejemplo de traducción automática basada en redes neuronales **28**

Figura 3.6 Representación del modelo de codificador-decodificador **29**

Figura 3.7 Arquitectura del modelo *Transformer* **30**

Figura 3.8 Fórmula de atención de producto escalado **32**

Figura 3.9 Arquitectura del codificador posicional del algoritmo *Transformer* **33**

Figura 3.10 Arquitectura del decodificador del algoritmo T*ransformer* **34**

CAPÍTULO IV

Figura 4.1 Perceptrón Multicapa 38

Figura 4.2 Ejemplo de red convolucional 1D para la clasificación de textos 40

Figura 4.3 Adaptación del modelo de Hofmann y Mehnert 43

Figura 4.4 Iteraciones en la traducción automática como proceso cíclico 44

CAPÍTULO V

Figura 5.1 Corpus paralelo del español al quechua 53

Figura 5.2 Código para la importación de los corpus para el idioma quechua y español 53

Figura 5.3 Corpus de los idiomas quechua y español, importado en el sistema 54

Figura 5.4 Código para la limpieza de los corpus, del idioma quechua y español 55

Figura 5.5 Corpus limpio del idioma quechua y español 55

Figura 5.6 Código de tokenización de los corpus del idioma quechua y español.....56

Figura 5.7 Creación del vocabulario de palabras del idioma quechua reducido a 4086 palabras y del español a 2737 palabras 56

Figura 5.8 Actualización de los vocabularios, en quechua que llegaría a conseguir 4088 palabras y en español 2739 palabras 57

Figura 5.9 Fórmula de decadencia de la tasa de aprendizaje 57

Figura 5.10 Proceso de entrenamiento de las redes neuronales.....58

Figura 5.11 Función *evaluate* que traduce frases de un idioma origen a tokens en un idioma destino 59

Figura 5.12 Matriz resultante del producto matricial QK^T, para obtener las similaridades más altas entre palabras del español y el quechua 61

Índice de Tablas

CAPÍTULO II

Tabla 2.1 Vocales del quechua boliviano.....11

Tabla 2.2 Consonantes del quechua boliviano.....12

Tabla 2.3 Pronombres personales del quechua boliviano 13

Tabla 2.4 Sufijos posesivos del quechua boliviano 13

Tabla 2.5 El alfabeto quechua.....15

CAPÍTULO III

Tabla 3.1 Desarrollo histórico de la traducción automática.....18

CAPÍTULO IV

Tabla 4.1 Escala de cinco puntos de LDC 47

CAPÍTULO VI

Tabla 6.1 Características del corpus usado 65

Tabla 6.2 Resultado de la medición de la traducción automática de palabras, del quechua al español 66

Tabla 6.3 Resultado de la medición de la traducción automática de palabras, del español al quechua.....67

Tabla 6.4 Resultado de medición de la traducción automática de frases, del quechua al español.....67

Tabla 6.5 Resultado de medición de la traducción automática de frases, del español al quechua 68

Tabla 6.6 Comparación de los resultados de medición del traductor automático del quechua al español y del español al quechua.....70

Capítulo I
Marco Introductorio

CAPÍTULO I
MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

El presente trabajo propone desarrollar un sistema de traducción español-quechua. El quechua es un idioma hablado por 1.613.210 habitantes de 4 años a más de edad en Bolivia (INE censo 2012). En la actualidad una persona hablante de esta lengua debe usar el español u otra lengua para poder acceder a la tecnología. Los entornos computacionales no proporcionan interfaces, correctores ortográficos o de estilo, ni contenidos en lenguas nativas. La aplicación del procesamiento de lenguaje natural (*NLP, Natural Language Processing*), a las lenguas originarias representaría un avance para incorporarlas al nuevo entorno tecnológico, y el desarrollo del NLP permitiría acercar este mundo a la vida cotidiana de los quechua hablantes.

El tema de la traducción es un problema complejo y por lo tanto un campo propicio para la investigación, por ello, en el marco teórico del presente trabajo incluye el procesamiento de lenguaje natural, que es un área de la ciencia de la computación que junto con el área de lenguajes de programación y compiladores han mejorado a lo largo de los años y se incluyen técnicas de reconocimiento de patrones. Todo lo anterior se combina con una modelación matemática – estadística adecuada al problema en particular.

En el diseño, análisis y recopilación de datos para el traductor automático, se aplicó el algoritmo *trasformer* con el modelo de traducción basado en redes neuronales, que son usados para la traducción automática en otras lenguas más usadas, como es el caso de la traducción inglés - español. Sin embargo, el presente trabajo se centra en plantear traducciones simples, como un primer paso, para la traducción de lenguas nativas.

A la conclusión del trabajo se evaluó el desempeño del traductor, siguiendo diferentes estrategias. Los resultados obtenidos son aceptables, considerando el corpus obtenido para este proyecto.

1.2. ANTECEDENTES

En la actualidad hay muchas aplicaciones similares entre ellas podemos mencionar 3 aplicaciones:

- En otra investigación, como la de los autores: **Alcaraz, N. A., & Alcaraz, P. A. (2020). Aplicación web de Análisis y Traducción Automática Guaraní–Español/Español–Guaraní,** tiene como objetivo desarrollar una aplicación web de análisis y traducción automática guaraní – español y español – guaraní para ello se necesitó diseñar un analizador léxico, sintáctico y semántico, elaborar los módulos necesarios para la aplicación web propuesta, implementar un esquema de traducción dirigida mediante los analizadores léxico, semántico y sintáctico, agregar una base de datos con un diccionario propuesto, integrar una interfaz sencilla a la aplicación web. Para dicho objetivo se utilizaron las herramientas WAMPSERVER con la base de datos MySQL y Eclipse para el lenguaje de programación JAVA. Para los analizadores se utilizó un algoritmo con enfoque de traducción basado en corpus, implementando traducción automática basada en ejemplos. Finalmente se logró desarrollar una aplicación web donde de acuerdo al usuario se puede obtener traducción automática de palabras, frases u oraciones del español al guaraní y del guaraní al español.

- Existe también, Apertium que es una plataforma de traducción automática libre y de código abierto que ha sido creada inicialmente para traducciones entre lenguas emparentadas, por: **Hugo David Calderón 1 , Vilca César David Mamani Calderón 2 , Flor Cagniy Cárdenas Mariño 3 & Edwin Fredy Mamani Calderón 4 (2009). Traductor Automático en línea del Español a Quechua, Basado en la Plataforma Libre y Código Abierto Apertium**, sin embargo por su evolución permite crear pares de lenguas divergentes. La implementación del traductor automático correspondió al quechua del este de Apurímac. El estudio se realizó durante el año 2013 en la región Apurímac-Perú. Las etapas en la traducción corresponden a la incubación del sistema traductor automático,

creación del diccionario monolingüe quechua y reutilización del diccionario monolingüe español, creación del diccionario bilingüe y creación de las reglas de transferencia estructural. El resultado del sistema de traducción automática presenta, más de 4000 palabras raíces, 5000 traducciones de palabras raíces entre español y quechua, reglas de transferencia estructural de quechua a español y reglas de transferencia estructural de español a quechua implementadas. Finalmente, la calidad del traductor automático aplicando el método WER, presentó un promedio de error de calidad de traducción de quechua a español de 19,48 y calidad de traducción de español a quechua con error de 24,19.

- Otro sistema de traducción que supone la automatización del proceso de traducción de palabras del castellano al quechua y viceversa, pertenece a los autores, **Indhira Castro Cavero, Jaime Farfán Madariaga (2007). Traductor morfológico del castellano y quechua**, que opera en tres fases (análisis, transferencia y generación) usando representaciones morfológicas para las palabras. Al traducir una palabra, el sistema no sólo devuelve la palabra convertida al otro idioma, sino también, muestra información lingüística de los componentes de la palabra.

1.3. PLANTEAMIENTO DEL PROBLEMA

La UNESCO identificó en el año 2010 que el cincuenta por ciento de las lenguas a nivel mundial se encuentran en peligro de desaparecer, seis mil lenguas son habladas únicamente por el cuatro por ciento de la población mundial y el noventa por ciento de las lenguas no están representadas en Internet (UNESCO 2010). Esto plantea un problema muy importante y trascendente para la cultura universal y los valores humanos, que no es exclusivo de nuestro país: la preservación de la cultura y las lenguas indígenas. Por lo tanto, cada vez que desaparece un lenguaje, la humanidad pierde una parte de su semántica universal y parte del patrimonio cultural.

La versión electrónica de la nueva edición del "Atlas UNESCO de las lenguas en peligro en el mundo" contiene datos actualizados de cerca de 2.500 idiomas de los seis mil

existentes, en particular el idioma quechua figura en este atlas elaborado por la Unesco, un documento cuya versión electrónica fue presentada en la ciudad de París (Francia) (UNESCO 2019).

Asimismo, la UNESCO catalogó a 30 lenguas nativas de Bolivia en nivel vulnerable o en peligro de extinción debido a que la mayoría de los niños hablan la lengua, pero su uso está restringido a determinados ámbitos como el hogar familiar, o en ciertos casos los niños ya no hablan la lengua originaria (UMSA 2018).

De acuerdo a los datos del INE, se puede extraer que existen 1.613.210 personas que hablan quechua en Bolivia, 604.626 hablantes en el sector urbanos y 1.008.584 en el sector rural, sin embargo la variación de auto-identificación con el idioma quechua de los censos 2001-2012, muestran un decrecimiento muy significativo, 1.555.641 auto-identificados con el idioma (censo del 2001) y 1.281.116 auto-identificados con el idioma (censo del 2012), lo cual arroja un tasa de decrecimiento del 21,43%, que indica que existe una disminución del nivel de auto-identificación con el idioma quechua en la población (INE, censo 2012).

En Bolivia, particularmente en el departamento de La Paz que cuenta con 61.349 hablantes del idioma quechua (INE, censo 2012), existe la falta de una enseñanza de la lengua escrita y por ende la escasa traducción del idioma español al idioma quechua, lo cual obstaculiza a sus hablantes poder acceder a textos en su propio idioma, en tanto al profesional bilingüe se ve obligado a formarse para llevar a cabo la castellanización forzada a las comunidades y esto afecta la supervivencia y futuro del idioma, y en cuanto a tecnologías en español como el Internet, se convierte en la aniquilación de los lenguajes que no son útiles, esto impulsa cada vez más la extinción de las lenguas poco habladas.

1.4. OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un sistema de traducción del idioma español al idioma quechua, el cual se basará en el Procesamiento de Lenguaje Natural (*NLP, Natural Language Processing*).

1.4.2 OBJETIVOS ESPECÍFICOS

- Construir un corpus de frases emparejadas entre español y quechua.
- Generar una herramienta para el pre procesado de datos, que nos permitirá la carga de ficheros al sistema.
- Desarrollar un tokenizador de texto, que ayudará a la herramienta traductora a predecir los resultados.
- Implementar el traductor, para la codificación y descodificación de palabras y el armado de frases.

1.5. JUSTIFICACIÓN

La comunicación es fundamental para la interacción en la sociedad y el lenguaje, es la principal vía por la cual se realiza esta comunicación y funge además como un factor para la unidad de los grupos étnicos, representando el papel de "lenguaje común, que es el pensamiento mismo y constituye un código compartido, un campo semántico elaborado históricamente, según el cual se organiza la comprensión del mundo" (Guillermo 1981). Este cuerpo semántico es propio y es el que hace que cada lengua guarde historia y conocimientos. Por lo tanto, cada vez que desaparece un lenguaje, la humanidad pierde una parte de su semántica universal y parte del patrimonio cultural inmaterial.

Son los pueblos originarios los que más han sufrido el impacto del uso de las Tecnologías de la Información y la Comunicación (TIC) ya que la transmisión de las lenguas indígenas es oral, de generación en generación. Algunas no cuentan con lenguaje escrito y no utilizan la tecnología o si lo hacen son afectadas por la obligación

de utilizar lenguas dominantes, como por ejemplo el español o el inglés (Martínez Casas 2011).

El trabajo propone una interacción fácil y eficiente mediante una herramienta automática de traducción de uso general, en las dos vías (español a quechua y quechua a español). Eso permite a los hablantes, de las dos lenguas, elegir los textos y los temas de su interés, intercambiando conocimientos de acuerdo a sus propias necesidades e intereses.

1.6. LÍMITES Y ALCANCES

1.6.1. Límites

El proyecto contempla las siguientes limitaciones:

- El corpus armado de los dos idiomas, contará con un número limitado de palabras, frases y oraciones.
- El sistema de traducción, se limitará a traducciones cortas y simples.
- El porcentaje de probabilidad de predecir una palabra dependerá de la cantidad de palabras dentro de un corpus, a más palabras mayor será la probabilidad de aciertos.

1.6.2. Alcances

El proyecto contempla los siguientes alcances:

- El sistema traducirá, según el dialecto del idioma con el que se haya armado el corpus.
- Las traducciones se realizan de manera veloz y óptima.
- La interacción al momento de realizar una traducción será de modo fácil y eficiente.

1.7. DISEÑO METODOLÓGICO

Para poder cumplir con los objetivos y metas se utiliza el método sistémico, con las siguientes etapas:

- **Investigación bibliográfica**

  Realizar una investigación de los diferentes algoritmos de traducción, los paradigmas existentes y las discusiones entre ellos para experimentar con los mejores algoritmos.

- **Recolección de materiales.**

  Tomando el planteamiento de que los algoritmos de traducción estadística y neuronal han generado los mejores resultados, se debe buscar una máxima cantidad de frases o textos en quechua emparejándose con el español en una base de datos o corpus. Cuanto mayor sea la cantidad de datos encontrados mayor será la probabilidad de generar buenas traducciones.

- **Programación y desarrollo del sistema.**

  Antes de poder utilizar el corpus emparejado, se necesita evaluar los textos recolectados y hacerlos aptos para su pre procesamiento posterior. La programación y desarrollo del sistema, se realizará usando un lenguaje de programación apto para realizar el proceso de entrenamiento del modelo de red neuronal que adoptemos, basado en el Procesamiento de Lenguaje Natural (NLP).

- **Análisis de los resultados.**

  Con base en los resultados obtenidos con las diversas estrategias de traducción, se generarán razonamientos previos, para mejorar el comportamiento del caso particular de los idiomas que se pretenden traducir.

Capítulo II
Idioma Quechua

CAPÍTULO II

IDIOMA QUECHUA

2.1. ANTECEDENTES

Quechua también denominada quichua es una familia de lenguas originaria de los Andes Centrales que se extiende por la parte occidental de Sudamérica. Es una macro lengua con una población hablante de más de 9'000,000 distribuidos en los países Perú, Argentina, Ecuador, Chile y Bolivia, es lengua co-oficial en Perú (SIL, 2013).

Diversas hipótesis sobre el origen y expansión de una de las lenguas indígenas más habladas del mundo: el quechua. Cerrón-Palomino (1987: 323-349) menciona las siguientes:

a) la hipótesis del origen serrano en la que se atribuye a Cuzco "como su centro inicial, y a las conquistas incaicas como su mecanismo de difusión, sostenida por Rowe (1950) y Riva Agüero ya en 1921.

b) la hipótesis de origen costeño postulada en 1911 por Manuel González de la Rosa, retomada por Porras Barrenechea en 1951 y fundamentada con los estudios dialectológicos y de reconstrucción que le dedicaron Parker y Alfredo Torero para el sustento de esta hipótesis concluyendo que los dialectos diferentes al cuzqueño correspondiente a la rama central peruana son más conservados, por tanto, son de mayor antigüedad.

c) la hipótesis del origen forestal o amazónico que asigna como foco de difusión entre Chachapoyas y Macas (ceja de selva) y fue sostenida por William H. Isbell en 1974 basándose en la arqueología y la ecología.

d) la hipótesis del origen ecuatoriano en la opinión de Tschudi, seguida por Middendorf, Brinton y Louisa Stark, hipótesis muy discutible que desde la lingüística no encuentra ningún asidero.

2.2. DESCRIPCIÓN LINGÜÍSTICA

Según Plaza (2005), la descripción lingüística, puede encubrir que la evolución histórica del quechua no es independiente de los fenómenos sociales, económicos, políticos del contexto en el que se desenvuelve. El Dr. Plaza fue promotor activo del proceso de normalización de la escritura de la lengua quechua, uno de cuyos resultados es el Diccionario de la Nación Quechua (2018), asimismo presenta la fonología, morfología y la escritura normalizada del quechua, que veremos a continuación.

2.2.1. Fonología

- **Vocales**

Al nivel fonémico, las vocales del quechua son tres: /i, u, a/. Al nivel fonético, hay una variación muy grande en los tres fonemas y según hablantes y variantes regionales. La tendencia predominante en Bolivia ha sido la caracterización de las vocales solamente incluyendo las variantes abiertas de las vocales cerradas, como se presenta esquemáticamente en la (tabla 2.1).

	Anteriores		**Centrales**		**Posteriores**
Cerradas	i				u
Medias		[e]		[o]	
Abiertas			a		

Tabla 2.1. Vocales del quechua boliviano.
Fuente: (Plaza, 2005: iv)

En la pronunciación opera, casi sin excepciones, una simple regla de distribución complementaria. Las vocales se actualizan fonéticamente como vocales medias (abiertas) si hay presencia de una consonante postvelar en la palabra.

puriq / [pureq] 'ir' *puka* 'rojo' *suqta* 'seis'

warmiqa / [warmeqa] 'mujer' *tuta* 'noche' *ch'uspi* 'mosca'

urqu / [orqo] 'montaña' *pisi* 'poco'

sunqu / sonqo ‘corazón’ *k’ullu* ‘madero’

qiru / *qero* ‘vaso’ *killa* ‘luna’

Es decir que las vocales /i, u/ se actualizan fonéticamente como [e, o] respectivamente, solo en presencia de una consonante postvelar; es decir, un caso claro de distribución complementaria.

- **Consonantes**

Las consonantes del quechua se caracterizan por la presencia de las postvelares y las laringales (oclusivas aspiradas y glotalizadas). El alfabeto único, promulgado por el Ministerio de Educación y Cultura, establece 25 fonemas consonantales para el quechua en Bolivia (ver tabla 2.2).

		Bilabiales	Dento-alveolares	Palatales	Velares	Postvelares
Oclusivas & africadas	Simples	p	t	ch	k	q
	Aspiradas	ph	th	chh	kh	qh
	Glotalizadas	p’	t’	ch’	k’	q’
Fricativas			s	(sh)	j	(x)
Laterales			l	ll		
Nasales		m	n	ñ		
Vibrante			r			
Semivocales		w		y		

Tabla 2.2. Consonantes del quechua boliviano.
Fuente: (Plaza, 1995: 58)

Las consonantes africadas, representadas en el cuadro por <ch>, <chh> y <ch’>, han sido ubicadas en la fila de las otras consonantes oclusivas con el fin de simplificar la presentación y no priorizar la descripción lingüística sobre la funcionalidad de la escritura. Además se incluyen las variantes fonéticas principales (léase, las que provocaban controversias en esos momentos) entre

paréntesis, indicando que no es necesario escribirlas. En el caso de las vocales, se indicaba que se podía escribir con tres o cinco vocales.

2.2.2. Morfología

- **Pronombres Personales**

En quechua los pronombres personales son tres: primera, segunda y tercera persona, que pueden aparecer en singular y plural. La primera persona plural, además, se ramifica en inclusivo y exclusivo. En la tabla 2.3, se presentan los pronombres, incluyendo a los plurales.

Singular	**Pronombre personal**	**Traducción**
1	ñuqa	'yo' (el hablante solo)
2	qam	'tú' (el oyente solo)
3	pay	'él, ella'
Plural		
1-Pl	ñuqa-nchik	'nosotros' (yo y vos-otros)
1-PE	ñuqa-yku	'nosotros' (yo y otros menos tú)
2-PL	qam-kuna	'ustedes'
3-PL	pay-kuna	'ellos, ellas'

Tabla 2.3. Pronombres personales del quechua boliviano.
Fuente: (Plaza, 2005: vi)

- **Sufijos posesivos**

Estos sufijos son los que denotan posesión o pertenencia tanto a personas, animales o cosas terminadas en vocal (ver tabla 2.4).

Singular	**Sufijos posesivos**	**Ejemplos**
1	-y	*p'anqa-y* 'mi libro', *chumpi-y* 'mi cinturón'
2	-yki	*tata-yki* 'tu padre', *chakra-yki* 'tu tierra'
3	-n	*ñaña-n* 'su hermana', *qullqi-n* 'su dinero'
Plural		
1-Pl	-nchik	*llaqta-nchik* 'nuestro país', *waka-nchik* 'nuestra vaca'
1-PE	-yku	*papa-yku* 'nuestra papa', *aqha-yku* 'nuestra agua'

2-PL	-ykichik	*wasi-ykichik* ‘vuestra casa’, *michi-ykichik* ‘vuestro gato’
3-PL	-nku	*chupa-nku* ‘sus colas’, *qhatu-nku* ‘su mercado’

Tabla 2.4. Sufijos posesivos del quechua boliviano.
Fuente: (Plaza, 2005: vi)

- **Sufijos pluralizadores**

Tradicionalmente el plural se marca con el sufijo *-kuna*. Quiróz (2000:71) dice: “que agregado a la palabra pluraliza en forma similar a la “s” del castellano”. Por ejemplo:

wasi-kuna ‘casas’

qhari-kuna ‘hombres’

allqu-kuna ‘perros’

p’isqu-kuna ‘aves’

2.2.3. Escritura normalizada

A partir de la publicación y uso de las cartillas autoinstructivas para el quechua, aimara y guaraní, se fueron difundiendo los principios de la escritura normalizada. En la cartilla *Qhichwata Qillqanapaq* (Plaza, 1995: iv-vi) se proponían, entre otros, los siguientes principios y reglas.

Los principios de la escritura normalizada son los siguientes: La escritura y la pronunciación son dos niveles de representación diferentes; la escritura no impide la pronunciación local o regional; la escritura no debe ser reducida a lo fonémico y local: la escritura al trascender los límites locales y fonémicos debe ser estudiada y aprendida.

Reglas para escribir el quechua: Las grafías entre paréntesis constituyen variantes de la pronunciación y no se utilizan en la escritura, normalizada. Para las vocales solamente hay necesidad de escribir las tres letras siguientes: <i>, <u>, <a>. Las letras <i>, y <u> responden a la regla de pronunciación siguiente:

i → [e] cuando hay /q/ en la palabra: *irqi* → [erqe]

→ [i] cuando no hay /q/ en la palabra: simi → [simi]

u → [o] cuando hay /q/ en la palabra: *urqu* → [orqo]

→ [u] cuando no hay /q/ en la palabra: *muju* → [muju]

Para las consonantes <ch>, <k>, y <q> opera una regla similar. Estas letras tienen dos pronunciaciones: una oclusiva delante de vocal y otra fricativa detrás de vocal en la sílaba.

ch → [ch] delante de vocal: *chunka* → [chunka]

[S] detrás de vocal: *phuch-ka* → [phuSka]

k → [k] delante de vocal: *kancha* → [kancha]

[x] detrás de vocal: *lliklla* → [llixlla]

q → [q] delante de vocal: *qallpa* → [qallpa]

[X] detrás de vocal: *llaqta* → [llaXta]

Algunos morfemas trascienden la escritura fonémica y requieren de una escritura basada en la morfología.

Orden alfabético de las letras quechuas:

a, ch, chh, ch', i, j, k, kh, k', l, ll, m, n, ñ, p, ph, p', q, qh, q', r, s, t, th, t', u, w, y

a	allqu	'perro'	p	puka	'rojo'
ch	chumpi	'faja'	ph	phuru	'pluma'
chh	chharpu	'borroso'	p'	p'akiy	'romper'
ch'	ch'iru	'astilla'	q	qam	'tu'
i	inti	'sol'	qh	qhaway	'mirar, cuidar'
j	jampi	'remedio'	q'	q'iru	'vaso'
k	kancha	'cancha'	r	rumi	'piedra'

kh	khituy	‘frotar’	s	saphi	‘semilla’
k’	k’antiy	‘entorcelar’	t	tapuy	‘preguntar’
l	layqa	‘brujo’	th	thuta	‘noche’
ll	llama	‘llama’	t’	t’usu	‘pantorrilla’
m	mama	‘mamá, madre’	u	urmay	‘caer’
n	nina	‘fuego’	w	wawa	‘criatura, niño’
ñ	ñawi	‘ojo’	y	yapa	‘aumento’

Tabla 2.5. El alfabeto quechua.
Fuente: (Plaza, 2005: vi)

Capítulo III
Estado del Arte

CAPÍTULO III

ESTADO DEL ARTE

El campo de conocimiento de la ciencia de la computación conocido como procesamiento de lenguaje natural permite el estudio de diversas tareas entre los lenguajes naturales y las computadoras, como son el reconocimiento de voz, la generación de textos, la extracción de información, la traducción automática, entre otras. Sin embargo, el NLP se ha centrado en los lenguajes más hablados y existen pocos ejemplos para lenguajes con pocos textos escritos. En el caso de la traducción automática casi no existen traductores automáticos para lenguas originarias, pero si han sido extensamente trabajadas para alemán, español, francés, italiano, portugués, árabe, japonés, coreano, chino, holandés, griego y ruso (en sistemas comerciales y públicos como Google, Systran, Prompt); y en casi todos los casos el inglés es la contra parte de las traducciones (Laukaitis & Vasilecas, 2007).

3.1. AVANCES TECNOLÓGICOS DE LA TRADUCCIÓN AUTOMATIZADA

Para poder entender el problema de traducción de una manera más amplia, se presenta un esbozo histórico de su desarrollo. En la tabla 3.1 se muestra una línea del tiempo de la traducción automatizada.

Año	**Suceso**	**Autor**	**Comentario**
1967	*Computational Analysis of Present-Day American English*	Henry Kucera y W. Nelson Francis	Este trabajo fue el puntapié para que el NLP comenzara a poner foco en el análisis de elementos lexicales y traducciones automáticas. Un corpus contenía aproximadamente un millón de palabras
1968	Systran	Peter Toma	Primer traductor basado en reglas comercializado (Inglés - Ruso)
1976	Météo	TAUM	Sistema de traducción de informes meteorológicos (Inglés-Francés)

1988	Traducción Estadística CANDIDE	Grupo de investigación IBM	Se presenta la traducción estadística en un modelo por palabras
1993	Verbomil	Ministerio Federal de Investigación alemán	Sistema de traducción simultánea basada en Interlingua
1999	Traducción por frases	Och, Tillman, Ney	Se plantea el primer modelo de traducción estadística por frases
2007	Inteligencia Artificial y el análisis de datos	*Google Translate*	Sistema de búsquedas en el navegador de internet y la solidez de *Google Translate*, usando NLP con un corpus de 40 mil millones de palabras
2012	Concurso en velocidad de traducción automática	Franz-Josef Och	*Google Translate* traducía suficientes textos como para llenar un millón de libros al día
2016	Tecnología de red neural	*Google Neural Machine Translation (NMT)*	Esta tecnología comenzó a traducir combinaciones de idiomas que no le habían enseñado

Tabla 3.1 Desarrollo histórico de la traducción automática.

En el inicio de la traducción automática se comenzaron a probar diferentes metodologías, que fueron desde una simple traducción directa, palabra a palabra, usando algunas reglas simples, hasta métodos más refinados que utilizaban análisis semántico y morfológico.

En la década de los 60 con los primeros análisis de texto mediante computadora. Un hito para el NLP y la lingüistica fue la publicación de *Computational Analysis of Present-Day American English* en 1967 de los autores Henry Kucera y W. Nelson Francis. En la publicación exponen cómo realizaron computacionalmente el primer corpus de NLP, *Brown Corpus of Standard American English*. Este corpus contenía aproximadamente un millón de palabras. Este trabajo fue el puntapié para que el NLP comenzara a poner foco en la extracción y análisis de elementos lexicales, en los métodos de estados finitos y en las primeras traducciones automáticas (Murzone, 2020).

Los primeros sistemas comerciales fueron presentados en los años 70, el primero en su tipo fue Systran, fundado en 1968 por Peter Toma. Fue usado desde 1970 por la Fuerza Aérea de los Estados Unidos y en un inicio únicamente traducía del ruso-inglés (Koehn, 2010). La Comisión Europea también adquirió una versión, esta vez inglés-francés, con lo que se comenzaron a desarrollar más pares de idiomas. Systran es un traductor basado en reglas y en la actualidad cuenta con cuarenta pares de idiomas, es multiplataforma y sigue desarrollándose. En 1976, se presentó el sistema MÉTÉO por el grupo TAUM (*Traduction Automatique de l'Université de Montréal*), desarrollado para traducir informes meteorológicos en Inglés-Francés y fue usado desde 1982 hasta el 2001.

En las décadas de los 80 y 90, surge el concepto de métodos de traducción impulsados por datos, con los primeros intentos realizados por traducción basada en ejemplos. En los laboratorios de IBM, surgió el modelo de una traducción estadística (Brown *et al.*, 1988), inspirándose en los métodos estadísticos de reconocimiento de voz que estaban dando sus primeros pasos. Sin embargo, en ese momento no tuvo mayores repercusiones, al estar el paradigma centrado en los sistemas basados en reglas e interlingua. El sistema que se desarrolló fue CANDIDE (Berger *et al.*, 1994), que fue el primer sistema estadístico basado en palabras. En 1998, los participantes en un taller de la Universidad de Johns Hopkins implementaron la mayoría de los modelos IBM (Brown *et al*., 1988) e hicieron públicas sus herramientas, lo cual permitiría la experimentación de más personas en el modelo, llevando a un rompimiento del paradigma imperante. Con los trabajos de Och, Tillman y Ney (Och *et al*., 1999) se comenzó la etapa de la traducción basada en frases.

El sistema más emblemático en software libre es Moses (Moses, 2016); pero también los traductores comerciales Bing y *Google Translate* funcionan con este paradigma. Desde entonces se ha trabajado en los dos principales paradigmas, la traducción estadística y la traducción basada en reglas, además de modelos híbridos entre los anteriores.

En los 2000 los traductores automáticos se hacen necesarios, logrando en la última década su uso frecuente, ya que fue una verdadera revolución tecnológica encabezada por la Inteligencia Artificial y el análisis de datos. La empresa Google sobresale no sólo por el sistema de búsquedas sino también por la solidez de *Google Translate*, usando CNTK y MSRLM, dos *toolkits* para NLP, que incorporan 40 mil millones de palabras. Recordemos el salto desde 1967 de 1 millón a 2007 de 40 mil millones y para no ser menos el NLP ha tenido los últimos años en los que mes a mes supera el *State of The Art* (SOTA) y por mucho, ya que siempre se sabe de algún algoritmo nuevo que permite crear un texto y traducir en *realtime* lo que decimos (Murzone, 2020).

Franz-Josef Och ganó un concurso en velocidad de traducción automática en el año 2003, y llegó a convertirse en director de Desarrollo de Traducción de Google. En el año 2012, Google anunció que su propia aplicación *Google Translate* traducía suficientes textos como para llenar un millón de libros al día. Japón también lidera la revolución de la traducción automática con la creación de traducciones de voz para teléfonos móviles, que funcionan en inglés, japonés y chino. Esto es el resultado de invertir tiempo y dinero en el desarrollo de sistemas informáticos con un modelo de red neuronal, en vez de funciones basadas en memorias. En el año 2016, Google aplicó la tecnología de red neural que mejoró la claridad de *Google Translate*, eliminando gran parte de su torpeza. La llamaron *Google Neural Machine Translation* (NMT). Esta tecnología comenzó a traducir combinaciones de idiomas que no le habían enseñado. Los programadores enseñaron al sistema inglés-portugués e inglés-español, y éste empezó a traducir portugués-español, a pesar de no haberle sido asignada esa combinación (Parra, 2020).

3.2. ANTECEDENTES DE TRADUCTORES AUTOMÁTICOS DEL ESPAÑOL AL QUECHUA

A través de la web podemos encontrar actualmente diferentes herramientas que ayudan a la traducción en el idioma quechua como ser: diccionarios on-line, glosarios terminológicos, diccionarios en la Web (diccionarios bilingües): sistemas que hacen uso

de una base de datos, en donde almacenan las palabras asociadas a su respectiva traducción.

Pero por otro lado en cuanto a traductores automáticos a quechua se refiere, existen muy pocos de los cuales podemos mencionar a dos:

3.2.1. Traductor automático en línea del español al quechua, basado en la plataforma libre y código abierto Apertium

Apertium es una plataforma de traducción automática libre y de código abierto que ha sido creado inicialmente para traducciones entre lenguas emparentadas, basado en sistemas de reglas de transferencia desarrollada por el grupo de investigación Transducens de la Universidad de Alicante de España. La arquitectura de Apertium usa transductores de estados finitos para el procesamiento léxico, modelos ocultos de Markov para la desambiguación léxica y procesamiento de patrones basado en estados finitos para la transferencia estructural (ver figura 3.1). Actualmente esta plataforma de traducción automática por transferencia ha permitido implementar y poner en marcha a más de 35 pares de lenguas como sistemas de traducción automática (Armentano *et al.*, 2007).

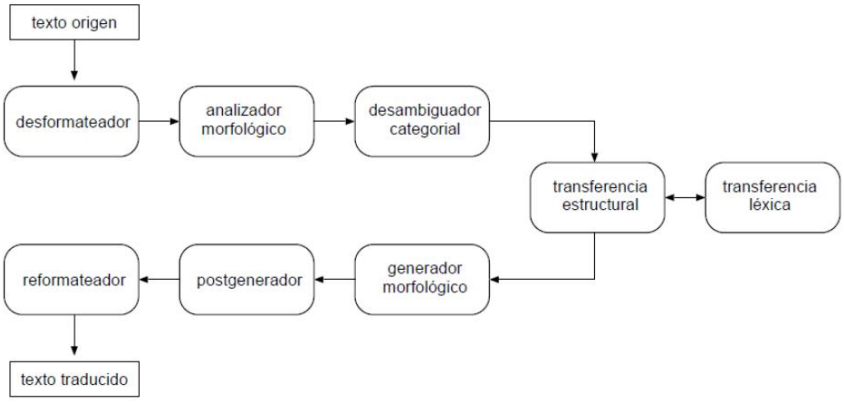


Figura 3.1. Módulos del sistema de traducción Automática de Apertium.
Fuente: (Armentano *et al.*, 2007)

Este traductor Automático implementa diccionarios morfológicos del idioma español y el diccionario morfológico de la lengua quechua, así mismo implementa el diccionario bilingüe del par español y quechua, de la misma manera define reglas de transferencia para la traducción del idioma español a la lengua quechua. En su evaluación, la calidad del Traductor Automático con la métrica WER (*Word Error Rate*), presentó un promedio de error de calidad de traducción de quechua a español de 19,48% y calidad de traducción de español a quechua con error de 24,19% (Calderon, 2009), siendo esta no tan aceptable para la comprensión de textos traducidos por el traductor automático.

3.2.2. Traductor morfológico del castellano al quechua (TECSUP)

Este sistema de traducción supone la automatización del proceso de traducción de palabras del castellano al quechua y viceversa, basado en la transferencia, que opera en tres fases (análisis, transferencia y generación) usando representaciones morfológicas para las palabras. Al traducir una palabra, el sistema no sólo devuelve la palabra convertida al otro idioma, sino también, muestra información lingüística de los componentes de la palabra (Castro y Farfán, 2004).

Figura 3.2. Proceso de traducción del traductor morfológico TECSUP.

Fuente: (Castro y Farfán, 2004)

Este traductor Automático desarrollado con herramientas tecnológicas de código abierto como Java, MySql y Apache, realiza una traducción palabra por palabra además que cumple un rol informativo, que trae como consecuencia el enriquecimiento lingüístico del idioma quechua, que resulta más explicativo que un simple inventario de términos, pero no cumple con la traducción entera de frases u oraciones y mucho menos de textos completos lo cual no cumple con las expectativas deseadas de un traductor automático completo.

3.3. MODELOS DE TRADUCCIÓN AUTOMÁTICA

Como se ha visto en la sección anterior, el desarrollo de la traducción automática (*Machine Translation*) se movió por momentos y tendencias. Se mostrara ahora los modelos más relevantes en la actualidad, los que se pueden dividir estas en tres paradigmas: la traducción basada en reglas RBMT, los modelos estadísticos SMT y la traducción basada en redes neuronales NMT (Parra, 2020). A continuación se explican estos modelos.

- **RBMT** (*Rule Based Machine Translation*) Comenzamos con este paradigma que, si bien sigue teniendo cabida en la investigación, es quizás el que menos atención acapara, aunque no por ello se deba obviar. Una de las posibles razones por las que este tipo de traducción automática no tiene más presencia en la investigación es porque requiere de una gran inversión en tiempo y recursos. Para desarrollar un único par de lenguas es necesario contar con gramáticas de la lengua de partida y de llegada, así como con diccionarios bilingües y reglas de transferencia. Además, el sistema no puede traducir estructuras lingüísticas que no estén incluidas en sus gramáticas o reglas de transferencia, ni palabras o expresiones que no se encuentren en sus diccionarios. Esto hace que el mantenimiento de los motores basados en reglas deba ser prácticamente continuo para garantizar que son capaces de traducir textos de un nuevo dominio, por

ejemplo, o estructuras gramaticales que en un primer momento no fueron previstas (Parra, 2020).

En el triángulo de Vauquois (Jurafsky & Martin, 2000), que se muestra en la figura 3.3, se explica cómo se logran relacionar diferentes niveles de reglas de traducción entre dos lenguajes.

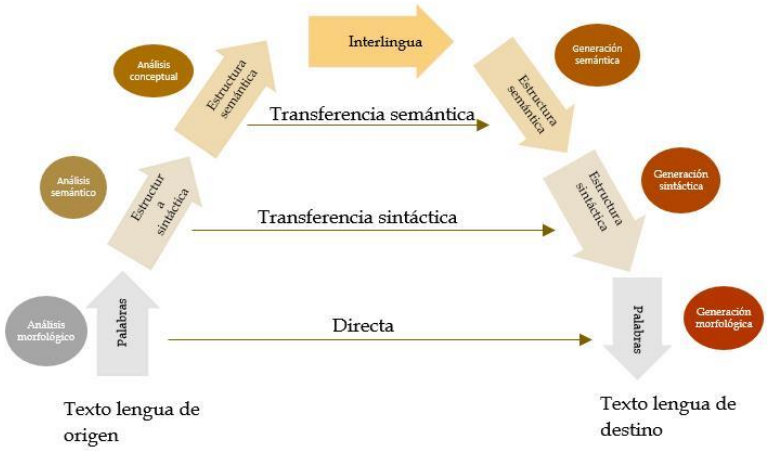
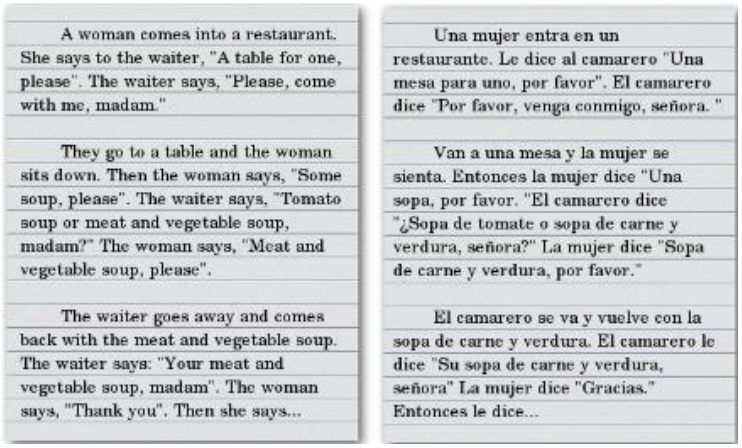


Figura 3.3. Triángulo de Vauquois.
Fuente: (Jurafsky & Martin, 2000)

- **SMT** (*Statistical Machine Translation*) Como alternativa a los costosos procesos de desarrollo de un traductor automático basado en reglas, se pueden emplear los llamados traductores automáticos estadísticos. Su principal ventaja frente a los basados en reglas es que tan solo necesitan datos para ser entrenados. En concreto, necesitan un corpus monolingüe de la lengua de destino lo más grande posible y otro paralelo con traducciones entre la lengua de origen y la de destino (ver figura 3.4).

  Estos sistemas de traducción automática constan de tres componentes principales: el modelo de lenguaje, el modelo de traducción y el decodificador. El modelo de lenguaje se encarga de calcular la probabilidad de que una frase en la lengua de destino sea correcta. Es el encargado de la fluidez de la traducción y para entrenarlo se utiliza un corpus monolingüe de la lengua de destino lo más grande posible. Por su parte, el modelo de traducción se encarga de establecer la correspondencia entre el idioma de origen y el idioma de destino y se entrena

utilizando un corpus alineado a nivel oracional. Durante esa fase de entrenamiento, el sistema estima la probabilidad de una traducción a partir de las traducciones que aparecen en el corpus de entrenamiento. Por último, el decodificador es el responsable de buscar dentro de todas las traducciones posibles la más probable en cada caso. Así, dado un modelo de lenguaje y un modelo de traducción, crea todas las traducciones posibles y propone la más probable (Parra, 2020).



A woman comes into a restaurant. She says to the waiter, "A table for one, please". The waiter says, "Please, come with me, madam."

They go to a table and the woman sits down. Then the woman says, "Some soup, please". The waiter says, "Tomato soup or meat and vegetable soup, madam?" The woman says, "Meat and vegetable soup, please".

The waiter goes away and comes back with the meat and vegetable soup. The waiter says: "Your meat and vegetable soup, madam". The woman says, "Thank you". Then she says...

Una mujer entra en un restaurante. Le dice al camarero "Una mesa para uno, por favor". El camarero dice "Por favor, venga conmigo, señora. "

Van a una mesa y la mujer se sienta. Entonces la mujer dice "Una sopa, por favor. "El camarero dice "¿Sopa de tomate o sopa de carne y verdura, señora?" La mujer dice "Sopa de carne y verdura, por favor."

El camarero se va y vuelve con la sopa de carne y verdura. El camarero le dice "Su sopa de carne y verdura, señora" La mujer dice "Gracias." Entonces le dice...

Lenguaje origen → Lenguaje objetivo

Figura 3.4. Ejemplo de oraciones en un corpus paralelo del inglés al español. Cada oración del inglés está alineada con su correspondiente traducción al español.
Fuente: (Parra, 2020)

- **NMT** (*Neural Machine Translation*) La última aparición en la familia de los paradigmas de traducción automática es la de los motores basados en redes neuronales en 2014 (Bahdanau, *et al.*, 2014). Se suele considerar como precursor de este paradigma un artículo publicado por dos investigadores españoles, Mikel Forcada y Ramón Ñeco, en el año 1997, en el que ya entonces proponían el uso de redes neuronales para la traducción automática. Si aquella idea no se llevó a la práctica fue, en parte, por una limitación muy clara: para poder llevar a cabo experimentos se necesitaban ordenadores y procesadores muy potentes, algo que

en esos años no estaba al alcance de cualquiera. Actualmente, sin embargo, son muchos los centros de investigación que ya cuentan con acceso a supercomputadores donde entrenar este tipo de motores de traducción automática. Y esto mismo ha fomentado su investigación, así como los grandes avances que se han conseguido, principalmente en los dos últimos años. En 2016 fueron precisamente estos motores los grandes ganadores en las competiciones que se organizan anualmente en el ámbito académico.

Al igual que los sistemas estadísticos de traducción automática, los sistemas basados en redes neuronales necesitan grandes corpus paralelos para su entrenamiento. De hecho, para que funcionen correctamente suelen necesitar que esos corpus paralelos sean aún más grandes que los que se necesitaban en el caso de la traducción automática estadística. Estas redes pretenden emular la manera en la que funcionan las neuronas en nuestro cerebro. Del mismo modo en que nuestras neuronas reciben información y realizan conexiones entre sí, los componentes del lenguaje se asocian con otra información subyacente para formar asociaciones y generar traducciones. Así, utilizando técnicas de aprendizaje automático, el ordenador aprende a traducir a partir de grandes cantidades de textos paralelos que además incluyen todo tipo de información lingüística y no lingüística (ver figura 3.5).

Cada palabra se convierte en sí misma más toda su información asociada, y esa información se utiliza para entrenar el motor. Gracias a la manera de relacionar la información asociada a cada palabra y a la de las palabras de una frase, el ordenador es capaz de aprender a traducir de una manera más eficiente (Parra, 2020).

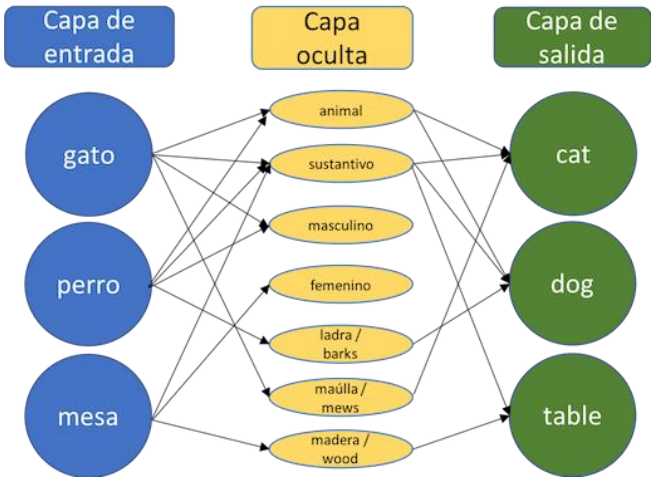


Figura 3.5. Los términos «perro» y «gato» son más similares entre sí que «perro» y «mesa», ya que en el texto generalmente estarán rodeados de palabras similares que nos pueden dar pistas sobre traducciones posibles.
Fuente: (Parra, 2020)

El potencial de la traducción automática basada en redes neuronales es tal que ya se están entrenando motores que, además de textos, incorporan imágenes o incluso archivos multimedia con resultados muy prometedores (Calixto *et al.*, 2017).

3.4. TRADUCCIÓN AUTOMÁTICA BASADA EN REDES NEURONALES (NMT)

En este trabajo se ha planteado tomar el camino del modelo NMT para el traductor español – quechua, como anteriormente explicamos, este modelo de traducción automática neuronal se hizo más poderosa y popular en los últimos años, pero de lo anterior, podemos deducir que NMT es un problema en el que procesamos una secuencia de entrada para producir una secuencia de salida, es decir, un problema de secuencia a secuencia (*seq to seq*).

Las propuestas iniciales se basaron en el uso de redes neuronales recurrentes RNN en una arquitectura de codificador-decodificador (ver figura 3.6). Estas arquitecturas tienen

una gran limitación cuando se trabaja con secuencias largas, su capacidad para retener información de los primeros elementos se pierde cuando se incorporan nuevos elementos a la secuencia. En el codificador, el estado oculto en cada paso está asociado con una determinada palabra en la oración de entrada, generalmente una de las más recientes. Por tanto, si el decodificador solo accede al último estado oculto del decodificador, perderá información relevante sobre los primeros elementos de la secuencia (Muñoz, 2020).

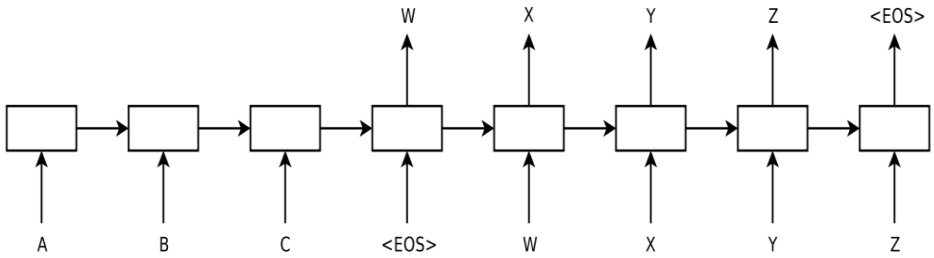


Figura 3.6. Representación del modelo de codificador-decodificador de Sutskever para la traducción de texto tomado de "Secuencia a secuencia de aprendizaje con redes neuronales".
Fuente: (Muñoz, 2020)

El punto crítico de este modelo es cómo hacer que el codificador proporcione la representación más completa y significativa de su secuencia de entrada en un solo elemento de salida al decodificador porque este vector o estado es la única información que el decodificador recibirá de la entrada para generar la salida correspondiente. Cuanto más larga sea la entrada, más difícil será comprimirla en un solo vector (Muñoz, 2020).

Por este singular problema de secuencia a secuencia, los investigadores comenzaron a explorar las capacidades de esta tecnología y se encontraron con nuevas soluciones, una de ellas es el uso del mecanismo de atención, siendo la principal mejora para un modelo denominado *Transformer*, uno de los modelos actuales más famosos que están surgiendo en tareas de NLP (Vaswani *et al.*, 2017).

3.4.1. Modelo Transformer

Según Muñoz (2020), el modelo *Transformer* extrae las características de cada palabra utilizando un mecanismo de auto-atención para descubrir qué tan importantes son todas las otras palabras en la oración para la palabra mencionada anteriormente. Y no se utilizan unidades recurrentes para obtener estas características, son solo sumas ponderadas y activaciones, por lo que pueden ser muy paralelizables y eficientes.

Pero profundizaremos en su arquitectura, para entender qué hacen todas estas piezas.

3.4.1.1. Arquitectura del modelo Transformer

En la arquitectura del modelo de traducción automática *Trasformer*, podemos observar que hay un modelo de codificador en el lado izquierdo y el descodificador en el derecho. Ambos contienen un bloque central de "una atención y una red de retroalimentación" repetida N veces, (ver figura 3.7).

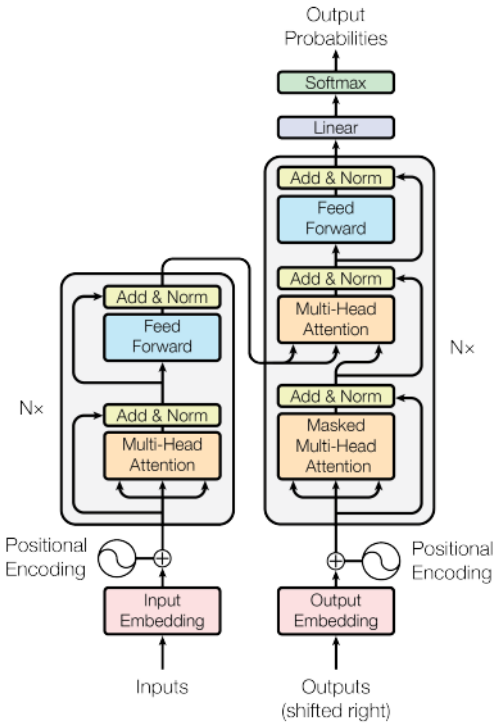


Figura 3.7. Arquitectura del modelo *Transformer.*
Fuente: "La atención es todo lo que necesitas" de Vaswani *et al.*, 2017.

Pero primero tenemos que explorar un concepto central en profundidad: el mecanismo de auto-atención.

- **Auto-Atención operación fundamental**

  La atención a uno mismo es una operación secuencia a secuencia: entra una secuencia de vectores y sale una secuencia de vectores. Vamos a llamar a los vectores de entrada *x1, x2,..., xt* y los correspondientes vectores de salida *y1, y2,..., yt.* Todos los vectores tienen dimensión *k*. Para producir un vector de salida *yi*, la operación de auto atención simplemente toma un promedio ponderado de todos los vectores de entrada, la opción más simple es el producto escalar (Bloem, 2019).

  En el mecanismo de auto-atención de nuestro modelo necesitamos introducir tres elementos: Consultas, Valores y Claves.

- **La atención del producto escalado**

  Según Vaswani *et al.,* (2017), la entrada consta de consultas y claves de dimensión *dk* y valores de dimensión dv. Calculamos el producto escalar de la consulta con todas las claves, dividimos cada una por la raíz cuadrada de *dk* y aplicamos una *softmax* función para obtener los pesos de los valores. Luego usamos las matrices *Q*, *K* y *V* para calcular los puntajes de atención. Los puntajes miden cuánto enfoque poner en otros lugares o palabras de la secuencia de entrada en una palabra en una posición determinada. Es decir, el producto escalar del vector de consulta con el vector clave de la palabra respectiva que estamos puntuando. Entonces, para la posición 1 calculamos el producto escalar: *q1* y *k1*, luego *q1. k2, q1. k3,* etc. A continuación, aplicamos el factor "escalado" para tener gradientes más estables. La función *softmax* no puede funcionar correctamente con valores grandes, lo que provoca la desaparición de los gradientes y ralentiza el aprendizaje. Después de "*softmaxing*", multiplicamos por la matriz de valores para mantener los valores de las palabras en las que queremos enfocarnos y minimizar o

eliminar los valores de las palabras irrelevantes (su valor en la matriz V debe ser muy pequeño). La fórmula para estas operaciones se muestra en la (figura 3.8).

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Figura 3.8. Fórmula de atención de producto escalado.
Fuente: "La atención es todo lo que necesitas" de Vaswani *et al.*, 2017

- **El codificador posicional**

  Se agrega la codificación de posición a la incrustación de entrada, (nuestras palabras de entrada se transforman en vectores de incrustación). La misma matriz de peso se comparte entre las dos capas de incrustación (codificador y decodificador) y la transformación lineal pre-softmax. En las capas de incrustación, multiplicamos esos pesos por la raíz cuadrada de la dimensión del modelo (Vaswani *et al.*, 2017).

  Existen 6 capas idénticas, que contienen dos subcapas: un mecanismo de auto-atención de múltiples cabezales y una red de alimentación hacia adelante completamente conectada (dos transformaciones lineales con una activación ReLU). Existe una conexión residual alrededor de cada subcapa (atención y red FC), que suma la salida de la capa con su entrada, seguida de una normalización de capa. Antes de cada conexión residual, se aplica una regularización: aplicamos el abandono a la salida de cada subcapa, antes de que se agregue a la entrada de la subcapa y se normalice. Además, aplicamos la deserción a las sumas de las incorporaciones y las codificaciones posicionales en las pilas de codificadores y decodificadores, con una tasa de deserción de 0,1. La normalización y las conexiones residuales son trucos estándar que se utilizan para ayudar a las redes neuronales profundas a entrenarse de forma más rápida y precisa. La normalización de la capa se aplica solo sobre la dimensión de incrustación (Bloem, 2019). La figura 3.9 muestra los componentes detallados.

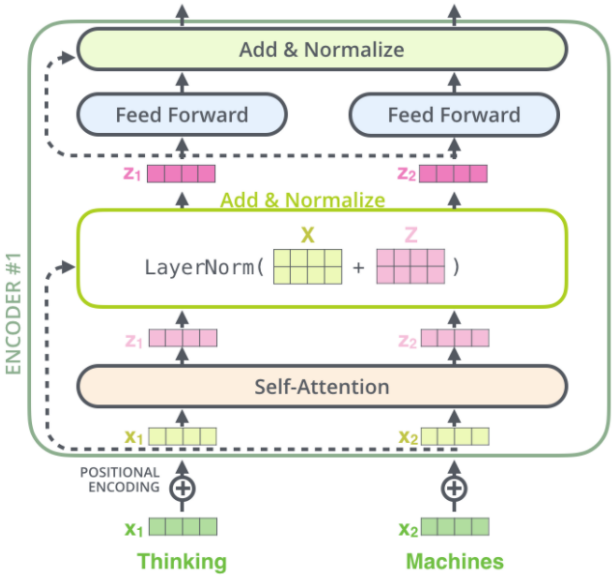


Figura 3.9. Arquitectura del codificador posicional del algoritmo *Transformer*.
Fuente: "El transformador ilustrado" de Alammar, 2018.

Tenga en cuenta que solo el vector de la última capa (6ª) se envía al decodificador.

- **El decodificador**

  Según (Vaswani *et al.*, 2017), el decodificador comparte algunos componentes con el codificador, pero se utilizan de forma diferente para tener en cuenta la salida del codificador, la codificación posicional es similar a la del codificador, tiene de igual manera 6 capas idénticas, pero esta vez contienen 3 tres subcapas. En primer lugar, la atención multicabezal enmascarada o la atención causal enmascarada para evitar que las posiciones atiendan a las siguientes.

  Este enmascaramiento, combinado con el hecho de que las incorporaciones de salida están compensadas por una posición, asegura que las predicciones para la posición i pueden depender solo de las salidas conocidas en posiciones menores que i. Se implementa configurando a $-\infty$ los valores correspondientes a los estados prohibidos en la capa softmax de los módulos de atención del producto escalar. El segundo componente o "atención del codificador-decodificador", realiza una atención de

múltiples cabezales sobre la salida del decodificador, los vectores de clave y valor provienen de la salida del codificador, pero las consultas provienen de la capa de decodificador anterior. Esto permite que cada posición en el decodificador atienda a todas las posiciones en la secuencia de entrada. Y finalmente la red este totalmente conectada. La conexión residual y la normalización de la capa alrededor de cada subcapa, es similar al codificador y repite la misma deserción residual que se ejecutó en el codificador. La figura 3.10 mostrará los componentes detallados.

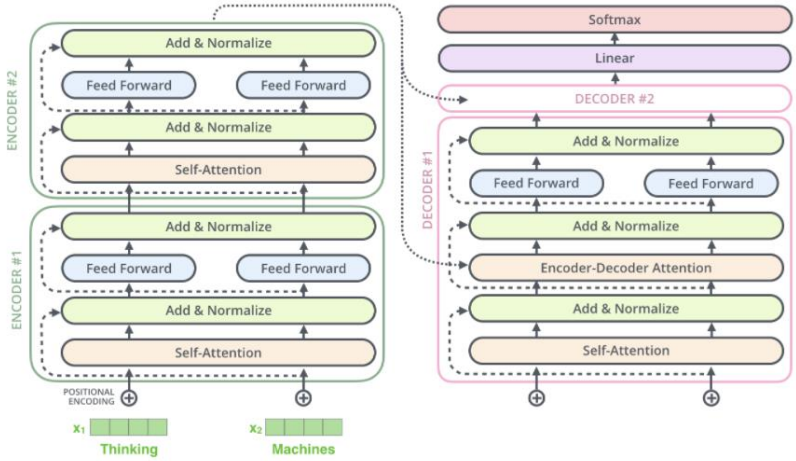


Figura 3.10. Arquitectura del decodificador del algoritmo T*ransformer*.
Fuente: "El transformador ilustrado" de Alammar, 2018.

Al final de los N decodificadores apilados, la capa lineal**,** una red completamente conectada, transforma las salidas apiladas en un vector mucho más grande, los logits. La capa softmax luego convierte esos puntajes (logits) en probabilidades (todos positivos, todos suman 1.0). Se elige la celda con la probabilidad más alta y la palabra asociada con ella se produce como la salida para este paso de tiempo (Alammar, 2018).

- **Uniendo todas las piezas del Transformer**

  Una vez que tenemos definidos todos los componentes y creado el codificador, el decodificador y la capa final linear-softmax, llamamos a todas estas para unirlas y asi obtener la salida predicha del modelo *Transformer*.

Capítulo IV
Marco Teórico

CAPÍTULO IV

MARCO TEÓRICO

4.1. PROCESAMIENTO DEL LENGUAJE NATURAL (NLP)

Según Murzone (2020), *Natural Language Processing* (NLP) es la rama del *Machine Learning,* dedicada a procesar el lenguaje. Se lo define como un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. El NLP se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio del lenguaje natural, es decir, de las lenguas del mundo.

El NLP se estudia desde la matemática y especialmente desde la estadística. Pero también entran en juego la lingüística y otras disciplinas. Es importante para comprender una comunicación estudiar la morfología de la misma, pero también su sintaxis, su semántica y su pragmática. Según la rama dentro del NLP puede ser conveniente estudiar la fonología humana o según el objetivo puede aportar al análisis del discurso.

4.1.1. Orígenes de NLP

El NLP surge aproximadamente en los 60’s con los primeros análisis de texto mediante computadora. Un hito para el NLP y la lingüística fue la publicación de *Computational Analysis of Present-Day American English* en 1967 de los autores Henry Kucera y W. Nelson Francis. En la publicación exponen cómo realizaron computacionalmente el primer corpus de NLP, *Brown Corpus of Standard American English*. Este corpus contenía aproximadamente un millón de palabras.

Este trabajo fue el puntapié para que el NLP comenzara a poner foco en la extracción y análisis de elementos lexicales, en los métodos de estados finitos y en las primeras traducciones automáticas.

En los 70’s se avanza en el análisis de texto y en la creación de técnicas lógicas y de razonamiento para extraer conocimiento de un texto.

En los 80’s la estadística entra con fuerza al campo para analizar las probabilidades basadas en pesos de que una palabra se encuentre en determinado lugar de la oración. Con esto aparecen también los primeros algoritmos de aprendizaje automático (*Machine Learning*) supervisados.

Los 90’s ya comienzan a anticipar una explosión en el campo. En primer lugar el crecimiento exponencial en el poder computacional y en la cantidad de datos almacenados a partir de la explosión de internet permiten extraer grandes cantidades de información de la red. Comienzan a desarrollarse algoritmos para reconocimiento de voz (*Speech to text o speech recognition*) y surgen algunas técnicas populares aún hoy como *Named Entity Recognition* (NER) o *Part-of-speech tagging.*

Los 2000 explotan al máximo lo logrado en la última década. Los buscadores web son los que se llevan los focos de las cámaras, Google se lleva todos los laureles. No sólo por el sistema de búsquedas sino también por la solidez de *Google Translate*. *Microsoft* hace lo suyo en aplicaciones del paquete office, particularmente en Word, reconociendo y recomendando con mucha precisión ya no sólo errores en la ortografía sino en la gramática.

4.2. REDES NEURONALES CONVOLUCIONALES (CNN)

Según Barrios (2020), las redes neuronales convolucionales (*Convolutional Neural Networks*) son un tipo de redes neuronales artificiales donde las “neuronas” corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria (V1) de un cerebro biológico. Este tipo de red es una variación de un perceptrón multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones.

Las redes neuronales convolucionales consisten en múltiples capas de filtros convolucionales de una o más dimensiones. Después de cada capa, por lo general se añade una función para realizar un mapeo causal no-lineal.

Dentro de la arquitectura total existe una primera capa, denominada como capa de entrada, la cual es la encargada de introducir al modelo las características de cada registro de nuestro conjunto de datos de muestra. De la capa de entrada se pasa a las capas intermedias, también llamadas capas ocultas, donde los distintos resultados a la salida de cada neurona pasan a las neuronas de las siguientes capas, hasta llegar a la capa final, conocida como capa de salida, la cual es la encargada de asignar una clase, en el caso de un problema de clasificación, o un valor numérico, en el caso de un problema de regresión.

La anterior descripción es básica, y refleja la morfología de la forma más simple de red neuronal artificial, el Perceptrón Multicapa (PMC), el cual es muy representativo a la hora de realizar una primera toma de contacto con las redes neuronales (ver figura 4.1).

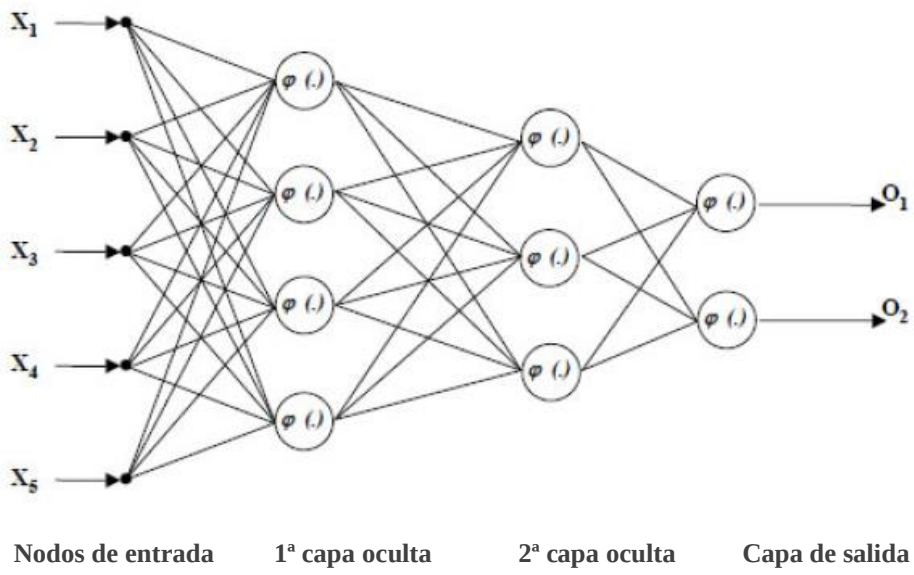


Figura 4.1. Perceptrón Multicapa.
Fuente: (Barrios, 2020)

El funcionamiento de las neuronas, también llamadas nodos, es sencillo. Los valores de salida de las neuronas de la anterior capa, o de las variables de entrada en el caso de que se trate de neuronas pertenecientes a la primera capa, son multiplicados por una serie de

valores, denominados pesos sinápticos, los cuales están asociados a cada conexión neurona-neurona. El resultado se suma teniendo un valor total que pasa a ser el input de la función de activación que caracteriza a la neurona. El resultado de la función de activación será la salida final de la neurona y esta a su vez pasará a las neuronas de la siguiente capa o será la salida del modelo, en el caso en el que las neuronas pertenezcan a la capa de salida.

4.2.1. Redes Neuronales Convolucionales para NLP

Según Barrios (2020), este tipo de redes, en su forma de 2D, son muy utilizadas en el camp de la visión artificial y características las convierten también en idóneas en su versión 1D para la clasificación de textos. Las redes convolucionales permiten que patrones, en este caso series de palabras, que se encuentran en una determinada posición puedan ser luego reconocidos en otra posición. La detección de estos patrones es lo que hará posible la correcta clasificación de los textos.

Previo a la entrada de los datos en lo que sería la red convolucional como tal, se realizan una serie de tratamientos, hasta llegar a la creación de los *Word Embeddings*. Al contrario que en la codificación de las palabras utilizando la metodología conocida como *one-hot encoding*, donde cada palabra se transforma en un vector de ceros con la dimensión del vocabulario total salvo por un uno en la posición que ocupe esa palabra en el diccionario, utilizando *word embeddings* podemos representar cada palabra por un vector, obteniendo representaciones similares para palabras con significados cercanos.

Pese a que existen distintos modelos pre-entrenados de *words enbeddings*, donde destacan *Word2Vec y GloVe*, para la resolución de este problema se creó una capa aleatoria de *words embedding* donde el modelo, al igual que recalcularía lo pesos de las neuronas de la red, recalcula los pesos de las distintas palabras dentro de un espacio determinado, de forma que el coseno del ángulo formado por los vectores que representan dos palabras similares sea próximo a cero, o lo que es lo mismo que estos vectores sean prácticamente iguales.

La salida de la capa de *words embedding* será una matriz con un número de filas igual al total de palabras de un texto y un número de columnas igual a la dimensión elegida para el vector de representación de cada palabra. Esta matriz será la entrada de la capa convolucional 1D. Sobre está matriz se extraerán subconjuntos de filas del tamaño elegido para el filtro de convolución, también llamado kernel, sobre los que se extraerán características hasta llegar a tener como resultado un vector de 1 dimensión el cual puede ser la entrada de una capa formada por neuronas con función de activación *softmax*. En la figura 4.2 se ilustra todo este proceso.

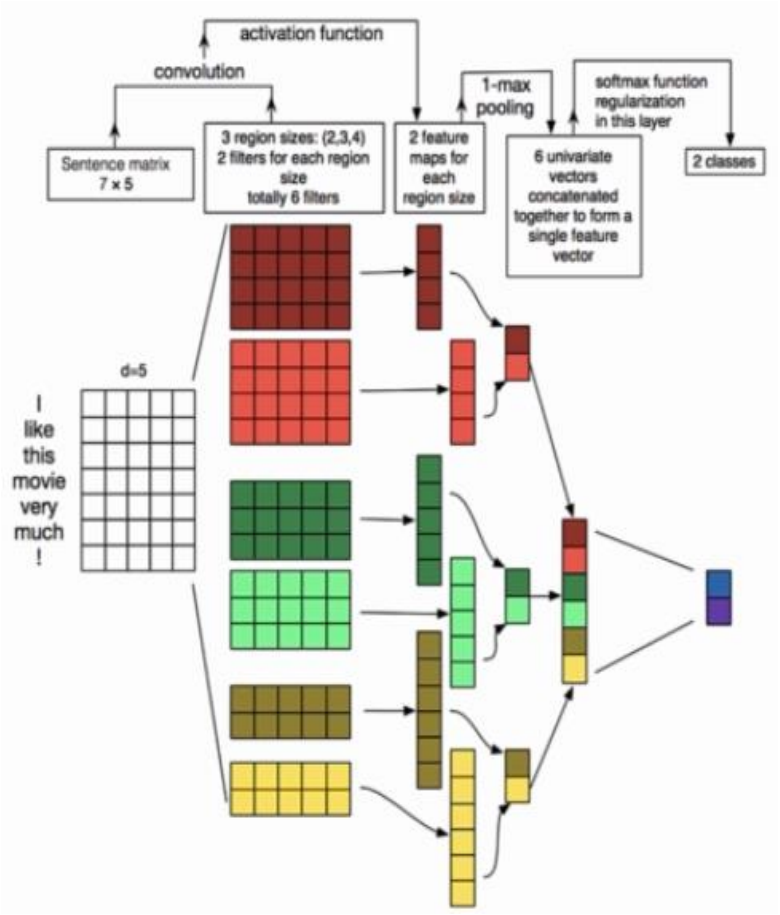


Figura 4.2. Red convolucional 1D para la clasificación de textos.
Fuente: (Zhang y Wallace, 2015).

4.3. LENGUAJE DE PROGRAMACIÓN PYTHON

Python es un lenguaje de programación versátil multiplataforma y multiparadigma que se destaca por su código legible y limpio. La licencia de código abierto permite su utilización en distintos contextos sin la necesidad de abonar por ello y se emplea en plataformas de alto tráfico como Google, YouTube o Facebook. Su objetivo es la automatización de procesos para ahorrar tanto complicaciones como tiempo, dichos procesos se reducen en pocas líneas de código que es compatible en una variedad de plataformas y sistemas operativos. Python se utiliza en prácticamente todas las industrias y campos científicos que pueda imaginarse, (Challenger *et al.*, 2014).

4.4. TENSORFLOW

TensorFlow es un marco de trabajo ampliamente utilizado para la inteligencia artificial. Google lo desarrolló y utilizó originalmente internamente, hasta que se lanzó como proyecto de código abierto en 2015. TensorFlow representa un cálculo de modelo como un modelo de flujo de datos en forma de gráfico dirigido. El gráfico se compone de un conjunto de nodos que representan operaciones, mientras que los bordes entre los nodos son tensores que contienen matrices de valores de dimensionalidad arbitraria. TensorFlow se basa principalmente en Eigen y cuBLAS como biblioteca para subrutinas de álgebra lineal subyacentes. En los dispositivos Android básicos, Eigen es la biblioteca que se utiliza. Mientras Eigenes una biblioteca muy bien optimizada para ejecutarse en procesadores ARM que utilizan el conjunto de instrucciones SIMD avanzado de ARM (NEON), no hace uso de otros recursos informáticos heterogéneos como GPU y DSP, (Goldsborough, 2016).

4.5. KERAS

Keras, es otro framework para trabajar con redes neuronales pero a más alto nivel, mediante APIs. Creada por François Chollet, un ingeniero de Google, es muy fácil de usar y permite una implementación muy rápida. Es una librería de código abierto que nació en 2015. Enfatiza el minimalismo por el hecho que puedes construir una red

neuronal con muy pocas líneas de código. Keras necesita un backend, un motor computacional que corra debajo, como Tensorflow, Caffe, Theano o CNTK (y otros). Podríamos poner el símil de un programa PHP que debe acceder a una base de datos. El programa no cambiará por acceder a MySQL, PostgreSQL o SQL Server. Keras sería el programa y Tensorflow, Caffe, etc la base de datos, (Ketkar, 2017).

4.6. METODOLOGÍA DE DESARROLLO

En las diferentes representaciones del proceso de traducción, tanto en el mundo académico (Gouadec 2007, Martín-Mor, Piqué y Sánchez-Gijón 2016, entre otros) como en el profesional (Dunne, 2011), se puede observar una tendencia a plantear los flujos de trabajo como herencia directa de los principios de la sociedad industrial, con una organización lineal de las tareas en la que hay una materia prima que entra en el sistema de producción (el texto original) y un producto que sale al final (el texto traducido). Desde esta perspectiva, se habla de alcanzar la máxima producción con el mínimo de costes, se propone una organización del trabajo en la que la máquina sustituye a la mano de obra humana y el empleo de la tecnología da lugar a la producción en masa para reducir costes y aumentar beneficios.

Sin embargo, cuando cambiamos el enfoque y miramos la traducción automática ya no como un producto aislado, sino como un instrumento con el que interactúa el usuario, en lugar de representar el proceso de traducción de manera lineal, se propone que lo visualicemos de manera circular. Esta idea ya en el inicio del siglo XXI Hofmann y Mehnert (2000: 63) introduce el concepto del ciclo de la información (*InfoCycle*) para explicar el proceso de gestión de la información multilingüe (Rico, 2017).

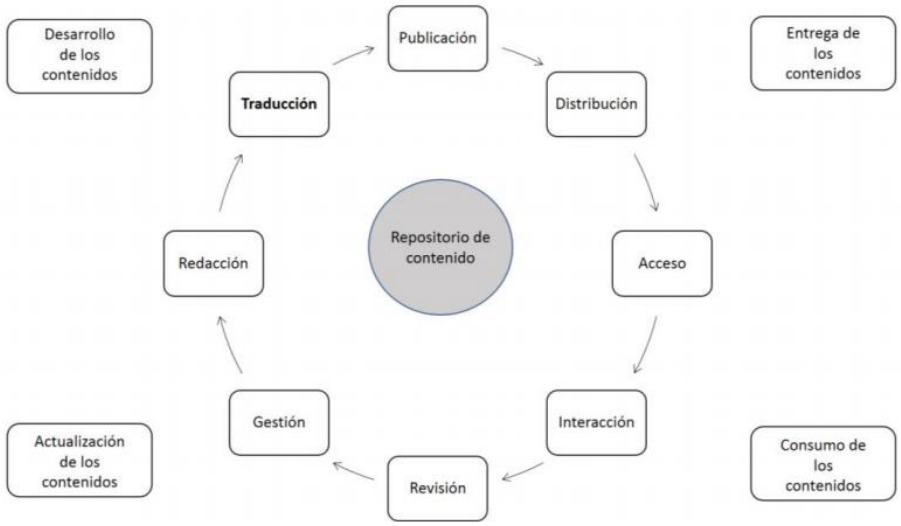


Figura 4.3. Adaptación del modelo de Hofmann y Mehnert (2000) sobre el ciclo de gestión de la información multilingüe.
Fuente: (Rico, 2017)

Según este modelo, la traducción es una parte indisoluble del ciclo de creación, revisión, publicación, distribución y consumo de la información multilingüe en un proceso continuo que se enriquece de la interacción con los diferentes actores que intervienen (redactores, traductores, gestores de proyecto, revisores y consumidor final). Es interesante mencionar que el planteamiento cíclico de los procesos está ya presente en otras disciplinas como el desarrollo de software, la gestión de recursos humanos o la definición de estrategias de liderazgo, en métodos como los Ágiles y Scrum (Álvarez y Gómez, 2011).

4.6.1. Proceso Cíclico

En esta metodología, el carácter cíclico del proceso es fundamental puesto que los proyectos se ejecutan en bloques temporales y cortos en diferentes iteraciones que siempre van acompañadas de una reflexión e interacción con el usuario. En cada una de las iteraciones, se debe proporcionar un resultado completo de manera que la entrega del producto final quede repartida en diferentes iteraciones y entregas, priorizadas según los objetivos del cliente. Cuando se adapta este modelo a la traducción automática y la redefinimos como un proceso cíclico, el enfoque se amplía y vemos que el traductor,

además de ser responsable de la entrega de un producto final de calidad, es también un actor clave en las diferentes etapas o iteraciones (Rico, 2017).

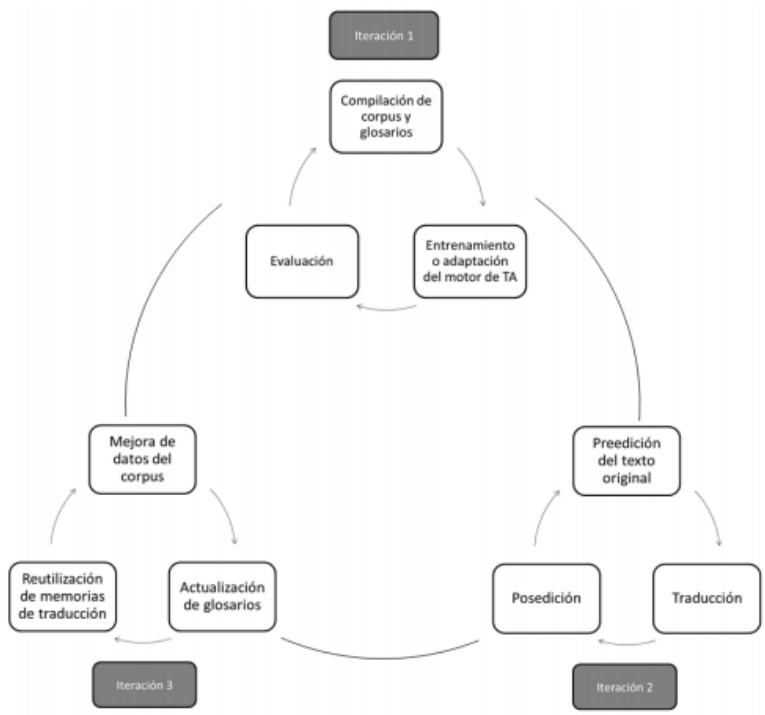


Figura 4.4: Iteraciones en la traducción automática como proceso cíclico.
Fuente: (Rico, 2017)

Veamos ahora estas interacciones en detalle junto con el papel que le corresponde al traductor y desarrollador en cada una de ellas.

Iteración 1

Esta primera etapa del proceso se centra en el propio sistema de traducción automática. Los hitos que la componen son los siguientes:

- **Compilación de datos y creación de glosarios específicos**. Estas son dos tareas esenciales en la implementación de cualquier sistema de traducción automática, tanto si se trata de motores de traducción neuronal, motores estadísticos o motores basados en reglas. En el primer y segundo caso, es necesario contar con

un corpus paralelo con el que se entrene el motor así como con uno monolingüe para crear el modelo de lengua. Para el tercer tipo de motores, la creación de glosarios específicos permite adaptarlos a las necesidades de cada cliente, texto o especialidad.

- **Entrenamiento de los motores de traducción automática.** La intervención del traductor en esta etapa es clave porque de ella depende el resultado final que se obtenga, esto es, un motor correctamente entrenado o adaptado generará traducciones de mayor calidad que otro en el que la adaptación es pobre o inexistente.
- **Evaluación del sistema de traducción automática.** En esta fase, el traductor debe determinar qué criterios de evaluación se van a usar y si estos son manuales, automáticos o, incluso, si se van a utilizar métodos que combinen ambas técnicas.

Iteración 2

Una vez completada la primera iteración, en la que el motor de traducción automática ya está operativo, la siguiente fase se centra en la traducción, desglosada del siguiente modo:

- **Pre edición del texto original**. Esta tarea tiene como objetivo la preparación del texto que se va a traducir de manera que se eliminen posibles fuentes de error cuando el sistema lo procese, por ejemplo, ambigüedades, frases subordinadas, empleo de la anáfora, términos desconocidos, entre otras cuestiones.
- **Traducción.** El motor de traducción automática origina una traducción de manera automática. Este es el único punto de todo el ciclo en el que el traductor humano no interviene.
- **Pos edición.** Se revisa el producto de la traducción automática según las reglas de pos edición que se hayan determinado previamente y en función de la calidad que se haya acordado con el cliente.

Iteración 3

- **Mejora de datos del corpus.** Esta etapa está encaminada a aumentar la calidad del corpus que se utiliza para entrenar el motor de traducción automática. Normalmente aquella se mide en función de la representatividad de los datos que lo componen, determinada a partir de la densidad léxica y en función del aumento incremental del corpus (Pastor y Domínguez, 2007).
- **Actualización de los glosarios**. Esta fase es importante para los sistemas de traducción automática basados en reglas puesto que, a diferencia de los estadísticos y neuronales, que no permiten la gestión terminológica (Pinnis *et al.*, 2014), la adaptación a cada especialidad pasa por el empleo de la terminología específica.
- **Reutilización de memorias de traducción** para volver a entrenar el motor de traducción con los textos ya traducidos (Moorkens *et al.*, 2014).

Esta metodología plantea pensar en la traducción automática como un ciclo iterativo que obliga a un cambio de enfoque en el papel que desempeña el traductor y le otorga un mayor protagonismo puesto que este deja de ser el "operario" que revisa un producto y se convierte en el "administrador" del proceso.

4.7. CRITERIOS DE EVALUACIÓN DE CALIDAD DEL TRADUCTOR AUTOMÁTICO

Podemos resumir que la evaluación de TA gira generalmente en torno a valorar si es lo suficientemente aceptable como para suponer un ahorro en la productividad de un traductor humano. La evolución en la evaluación de TA ha ido siempre a la par del desarrollo de las tecnologías de TA. Durante los inicios de la evaluación de TA, se recurría necesariamente a evaluadores humanos. Sin embargo, el uso de jueces humanos traía consigo ciertos problemas como la subjetividad, el coste y el tiempo (Kit & Wong, 2015: 223). En un intento de superar estas cuestiones, se desarrollaron métricas de evaluación automáticas como las que listaremos a continuación. Parece, por tanto,

previsible que en un futuro cercano se evalúe la TA exclusivamente con métodos automáticos.

4.7.1. Evaluación humana

Aunque este tipo de evaluación se considera a menudo subjetiva o inconsistente, son, sin embargo, el estándar de oro al que aspiran las métricas automáticas. Habitualmente, la evaluación se diseña de tal forma que unos evaluadores humanos puntúan si una traducción es buena o no, según su juicio, habitualmente segmento a segmento. Los criterios de evaluación más usados son fidelidad y comprensibilidad. Fidelidad consiste en saber si el significado del texto original que se ha transferido al texto traducido es preciso y adecuado, sin pérdidas, adiciones o distorsiones. Por otra parte, la comprensibilidad es una característica exclusiva del texto traducido y se refiere a la facilidad con la que se entiende una traducción. Ambos criterios se puntúan según una escala (Saldanha & O'Brien, 2014: 102). El uso de escalas en evaluación de la TA es muy habitual, como la de cinco puntos que se muestra en la siguiente tabla.

ESCALA	FIDELIDAD	COMPRENSIBILIDAD
5	Todo	Impecable
4	Sumamente	Bueno
3	Mucho	No nativo
2	Poco	Difluente
1	Ninguna	Incomprensible

Tabla 4.1. Escala de cinco puntos de LDC (2002)
Fuente: (Rico, 2017)

Está considerado como un método más fiable porque identificar errores es más objetivo y consistente que puntuar una traducción. Otro argumento a favor es que los resultados que proporciona este modelo son de mayor utilidad y más comprensibles para las partes interesadas como desarrolladores de sistemas o usuarios (Kit & Wong, 2015: 223).

4.7.2. Métodos automáticos

La evaluación automática de resultados de TA implica usar métricas cuantitativas sin intervención humana durante la ejecución del proceso. Está pensada para superar las deficiencias de la evaluación humana, esto es, por un lado, la subjetividad e inconsistencia, y, por el otro, el coste monetario y de tiempo. «*Automatic metrics serve thus as a desirable solution providing a quick and cost-effective means for trustable estimation of the quality of MT output*» (Kit & Wong, 2015: 225).

Dentro de las distintas métricas automáticas existentes, se puede diferenciar entre BLEU, NIST, METEOR, TER y ATEC. A continuación, describiremos las métricas más empleadas en la industria:

- **BLEU** (Papineri *et al.,* 2001) es la métrica más usada y se basa en la premisa de que cuanto más se acerque la traducción automática a la traducción humana, mejor es. Calcula el número de n-gramas (secuencia de palabra(s) consecutiva(s) de distinta longitud) que coinciden de la traducción en bruto con respecto a una o más traducciones de referencia. La TA debe coincidir con la traducción de referencia en la selección de palabras, en el orden de palabras y en la longitud para obtener un buen resultado. Por tanto, no se compara el resultado de la TA con el texto original, si no con una traducción de referencia. El resultado es siempre mayor o igual que cero y menor o igual que uno. Cuanto más se acerque al uno, de mejor calidad se considerará.
- **TER** (Snover *et al.,* 2006: 223-231) es una métrica de evaluación basada en la cuantificación de la distancia de edición entre dos cadenas de texto. Cuanto menor sea el número de operaciones requeridas para transformar una cadena en la otra, mejor será la evaluación. Se puede usar para medir el esfuerzo de posedición necesario para convertir un candidato en referencia. TER se formula como el número mínimo de inserciones (INT), eliminaciones (DEL), sustituciones (SUB) y cambios (SHIFT) de palabras necesarios para cada candidato.

- **NIST** (Doddington, 2002: 138-145) nace como una revisión de la métrica BLEU. Mientras que BLEU pondera todos los n-gramas de la misma forma, NIST le da un mayor peso a los n-gramas que son informativos. Cuanto menos se repita un n-grama en la traducción de referencia, más informativo se considerará.

Capítulo V
Marco Aplicativo

CAPÍTULO V

MARCO APLICATIVO

En este capítulo se describe paso a paso, el desarrollo del prototipo de traducción automática de la lengua quechua al español y viceversa. El corpus cuenta con 1.919 frases, los que fueron utilizados en el proceso de entrenamiento de la red neuronal NMT.

5.1. ANÁLISIS PARA EL DESARROLLO DEL PROTOTIPO

Los componentes del prototipo que identificamos son: los dialectos del idioma quechua, los lenguajes de programación, machine learning y la red neuronal.

- **Dialectos del idioma quechua:** al conocer la existencia de una gran variedad de dialectos en el idioma quechua, se optó por realizar la construcción del corpus en quechua, de textos escritos pertenecientes a la familia quechua conocida como Cuzco-Collao que a su vez es parte del Quechua Sureño (Quechua II-C), dependiente del ramal Chinchay del Quechua II. Cabe añadir que entre el quechua de Cuzco y el quechua boliviano hay cierta comprensión mutua (Alfredo Torero 1964), a igual de los dialectos que existen entre el quechua del norte paceño, cochabambino, sucrense y potosino.

- **Lenguaje de programación:** el lenguaje de programación utilizado fue Python en su versión 3.6.8, por ser un lenguaje sencillo, rápido, liviano e ideal para trabajar con machine learning y redes neuronales.

- **Machine learning:** para el procesamiento del lenguaje natural (NLP) se requiere una serie de herramientas de aprendizaje automático, por lo que se usó las librerías de Tensorflow en la versión 2.0.0, y Keras en la versión 2.2.4-tf.

- **Red neuronal:** entre los distintos tipos de redes neuronales que pueden implementarse, se decidió implementar las redes neuronales tipo *transformer,*

son una clase reciente de redes neuronales para secuencias, basadas en la auto atención, que han demostrado estar bien adaptadas al texto y actualmente están impulsando importantes avances en el procesamiento del lenguaje natural.

5.2. DISEÑO Y MODELACIÓN DEL TRADUCTOR AUTOMÁTICO

El modelo de traducción basado en redes neuronales NMT propuesto, esta implementado en la arquitectura del algoritmo *Transformer*, el cual utilizamos en el presente trabajo, este algoritmo puede realizar las tareas de NLP utilizando los bloques de codificación y descodificación con sus respectivas capas de atención, redes neuronales y normalización, a estas herramientas se pre procesa la entrada del corpus en español y quechua.

Para el desarrollo del traductor automático español – quechua, adoptamos la metodología propuesta en el anterior capítulo de marco teórico, que se basa en un proceso de carácter cíclico el cual nos recomienda dividir el proceso de desarrollo en iteraciones y estas a la vez de forma circular.

5.2.1. ITERACIÓN 1

Esta primera etapa del proceso se centra en el propio sistema de traducción automática.

5.2.1.1. Compilación de datos

Previamente, se requieren los datos a utilizar para poder desarrollar el traductor español - quechua, para ello se utilizó un corpus paralelo bilingüe, para traducir palabras, frases u oraciones de un idioma a otro. Con este fin se recopiló una serie de frases y textos quechuas disponibles en diferentes fuentes (Ramirez, 2017, Acuña, 2020), una muestra de estos se observa en la figura 5.1, que cuenta con 1.919 frases alineadas en ambos idiomas, 22.040 palabras en español y 12.630 palabras en quechua.

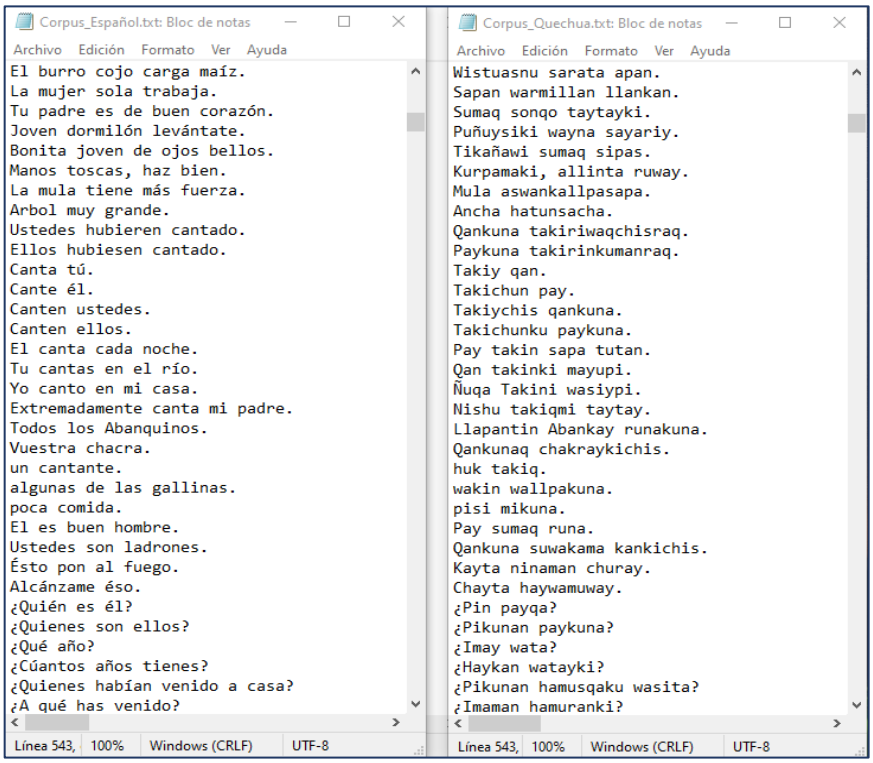


Figura 5.1: Corpus paralelo del español al quechua. Cada oración del español está alineada con su correspondiente traducción al quechua.

- **Carga de Ficheros**

  En esta fase, con los corpus ya alineados paralelamente, se realizó la importación de los datos, cargando estos dos corpus que van a hacer falta para poder implementar el algoritmo de *Transformer*. Para ello se llevó a cabo las dos cargas respectivas de los ficheros para abrirlos, en la figura 5.2 podemos observar el código de importación de datos y en la figura 5.3 observamos los corpus, tanto del idioma quechua como del español ya importados en el sistema.

```
In [3]: with open("corpus español-quechua/corpus_quechua.qv",
                   mode = "r", encoding = "utf-8") as f:
            quechua_qv = f.read()
        with open("corpus español-quechua/corpus_español.es",
                   mode = "r", encoding = "utf-8") as f:
            español_es = f.read()
```

Figura 5.2: Código para la importación de los corpus para el idioma quechua y español, en el sistema.

Figura 5.3: Corpus de los idiomas quechua y español, importado en el sistema.

- **Limpieza de datos**

  La limpieza de datos, *data cleansing o scrubbing* es un proceso necesario para asegurar la calidad de los datos, este paso es fundamental para eliminar la información poco precisa, errónea o incompleta.

  Existen varias funciones para eliminar errores de los datos, pero para este trabajo solo se utilizó las siguientes tres tareas con las funciones de limpieza de datos:

  - Recortar valores de cadena.
  - Reemplazar caracteres de una cadena.
  - Buscar patrones de coincidencia en expresiones regulares.

  Estas funciones se las implementó en el sistema mediante el código que se muestra en la figura 5.4, una vez implementadas estas funciones, los corpus

quedan limpios de espacios múltiples, frases sin fin y cada corpus quedaría separado por frases, como se ve en la (figura 5.5).

```
In [9]: corpus_qv = quechua_qv
        # Añadimos $$$ después de los puntos de frases sin fin
        for prefix in non_breaking_prefix_qv:
            corpus_qv = corpus_qv.replace(prefix, prefix + '$$$')
        corpus_qv = re.sub(r"\.(?=[0-9]|[a-z]|[A-Z])", ".$$$", corpus_qv)
        # Eliminamos los marcadores $$$
        corpus_qv = re.sub(r"\.\$\$\$", '', corpus_qv)
        # Eliminamos espacios múltiples
        corpus_qv = re.sub(r"  +", " ", corpus_qv)
        corpus_qv = corpus_qv.split('\n')

        corpus_es = español_es
        for prefix in non_breaking_prefix_es:
            corpus_es = corpus_es.replace(prefix, prefix + '$$$')
        corpus_es = re.sub(r"\.(?=[0-9]|[a-z]|[A-Z])", ".$$$", corpus_es)
        corpus_es = re.sub(r"\.\$\$\$", '', corpus_es)
        corpus_es = re.sub(r"  +", " ", corpus_es)
        corpus_es = corpus_es.split('\n')
```

Figura 5.4: Código para la limpieza de los corpus, del idioma quechua y español.

```
In [16]: corpus_qv
Out[16]: ['Santa Clara wamrakunapa cuentunkuna.',
          'Atoqwan usha.',
          'Uk atoqshi purikänaq allqapa. ',
          'Chayćhöshi üshata qewata mikĭkaqta tarinaq. ',
          'Atoqshi üshata manchakächinaq. ',
          'Chayshi ayqikur üshaqa aywakunaq. ',
          'Atoqshi üshapa qepanta cörrir aywanaq charipänanyoq. ',
          'Chayćhöshi michikoq wamra atoqta rikaskir qayaränaq: “¡Ama üshätaqa!” Niykaptinpis ashkallanta mikuskinaq. ',
          'Chayshi mamanqa wamranta tapunaq: “¿Maytaq ashkallanchikeqa?” “Atoqmi sugapumashqa” nir waqakunaq wamraqa. ',
          'Chayshi maman wamranta shoqanaq: “Ama waqëku. Awkamï kayman puñukushunpaq” nir.',
          '¿Maypiraq mamay kachkan?',
          'Ischu ukupis huk yutu uqllasqa, chay yutus nisqa: "trigo maskakuq risaq", nispa.',
          'Chaysi uqllasqanmanta trigo maskakuq risqa, kunallanmi kutiramusaq nispansi, chay yutu risqa.',
          'Qichpachallantas risqa trigo maskakuq, hinaspas manaña uqllasqañachu.',
          'Mana maman uqallachkaptinsi huk yutucha tuqyaramusqa.',
          'Chay yutuchas qawasqa waqtankunata, hinaspas mana rikusqachu mamanta. ',
          'Chay tuqyaramuq yutuchas sunqunta tapukusqa: ¿maypitaq mamay kachkan?, nispa. ',
          'Manas mayqin waqtapipas kasqachu, hinaspas chay yutucha nisqa: risaqyá mamay maskaq, nispa.',
          'Yutuchas maman maskaq risqa. ',
          'Manas yachakusqachu imayna kasqanta, mana maman uqllachkaptinsi tuqyaramusqa. ',

In [15]: corpus_es
Out[15]: ['Cuentos de los niños de Santa Clara. ',
          'El zorro y la oveja.',
          'Un zorro estaba andando por la puna. ',
          'Allí encontró una oveja comiendo pasto. ',
          'El zorro asustó a la oveja. ',
          'Entonces la oveja se fue escapando. ',
          'El zorro se fue corriendo detrás de la oveja hasta alcanzarla. ',
          'Un niño pastor vio al zorro allí y gritó: “¡A mi oveja no!” Aunque dijo eso, el zorro comió su corderito. ',
          'Luego su mamá preguntó al niño: “¿Dónde está nuestro corderito?” “El zorro me lo robó”, dijo el niño llorando. ',
          'Entonces la mamá acarició a su niño: “No llores. Ven acá para que durmamos”, dijo.',
          '¿Dónde estará mi madre?',
          'Dicen que dentro de la paja una perdíz empollaba, dice que esa perdíz decía: "voy en busca de trigo". ',
          'Así dice que de lo que empollaba se fue en busca de trigo, regreso pronto diciendo esa perdiz se fue. ',
          'Dice que se fue muy cerca en busca de trigo, y así ya no empollaba.',
          'Cuando la madre no estaba empollando un polluelo salió del cascaron. ',
          'Esa pequeña perdíz veía a su alrededor y así dice que no halló a su madre. ',
          'Dice que ese polluelo que salió se preguntó: ¿dónde está mi madre? diciendo. ',
          'Dice que no estaba por ningún lado, dice que entonces se dijo: iré en busca de mi madre, diciendo.',
          'Dice que el polluelo se fue en busca de su madre. ',
          'Dice que no sabía cómo era, había nacido sin que lo estuviera empollando. ',
```

Figura 5.5: Corpus limpio del idioma quechua y español, separado por frases y sin espacios múltiples.

5.2.1.2. Proceso de entrenamiento del motor del traductor automático

El proceso de entrenamiento requiere el uso del corpus paralelo español - quechua, con las frases emparejadas, tal que puedan ser analizadas como equivalentes. Para entrenar el modelo de la red neuronal tipo *transformer*, se creó un vocabulario de 2739 palabras en español y 4088 en quechua, asociados a sus respectivos tokens.

El proceso de tokenización, consiste en transformar cada palabra o conjunto de caracteres en un número, para que la red neuronal entienda esos números y pueda llevar a cabo el cálculo correcto. En el capítulo del estado del arte pudimos observar el funcionamiento de estas redes neuronales, que utiliza productos de matrices, por tanto, a la red neuronal le da absolutamente igual si la palabra es "andar", "caminar" o "correr", lo que requiere es el número que corresponde a cada una de esas palabras. Para ello se tokenizó a partir de los corpus, como se ve en la (figura 5.6).

```
tokenizer_qv = tfds.features.text.SubwordTextEncoder.build_from_corpus(
    corpus_qv, target_vocab_size=2**13)
tokenizer_es = tfds.features.text.SubwordTextEncoder.build_from_corpus(
    corpus_es, target_vocab_size=2**13)
```

Figura 5.6: Código de tokenización de los corpus del idioma quechua y español.

Una vez que ya tuvimos los textos tokenizados, se creó dos variables globales como se ve en la figura 5.7, donde se almacenara el tamaño del vocabulario en español y el tamaño del vocabulario en quechua.

```
tokenizer_qv
```

```
<SubwordTextEncoder vocab_size=4086>
```

```
tokenizer_es
```

```
<SubwordTextEncoder vocab_size=2737>
```

Figura 5.7: Creación del vocabulario de palabras del idioma quechua reducido a 4086 palabras y del español a 2737 palabras.

También, adicionamos dos tokens a cada uno de estas variables globales, dos tokens especiales, un token que indica inicio de frase y otro que indicará final de frase, esta adición se puede observar en la (figura 5.8). Cabe recalcar que estos tokens adicionales, no van a ser palabras de verdad que vaya a utilizar el modelo para entrenar, pero sí es importante considerarlas como parte del vocabulario con el que se trabaja.

```
VOCAB_SIZE_QV = tokenizer_qv.vocab_size + 2 # = 4088
VOCAB_SIZE_ES = tokenizer_es.vocab_size + 2 # = 2739
```

Figura 5.8: Actualización de los vocabularios, en quechua que llegaría a conseguir 4088 palabras y en español 2739 palabras sumándole estos dos tokens a cada variable global.

Posteriormente una vez culminada la toquenización, pasamos a compilar las redes neuronales mediante la librería Keras. Para ello creamos un generador de datos por lotes, para entrenar en lotes, creamos también una función de pérdida personalizada para enmascarar los tokens de relleno. Utilizamos el optimizador Adam, con beta1=0.9, eta2=0.98 y épsilon=10e-9. Y finalmente creamos un programador para variar la tasa de aprendizaje durante el proceso de capacitación de acuerdo con la siguiente fórmula:

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

Figura 5.9: Fórmula de decadencia de la tasa de aprendizaje.
Fuente: “Atención es todo lo que necesitas” de Vaswani, *et al.*, 2017.

A medida que se ejecute la red neuronal, observamos que el valor de pérdida baja con cada iteración y en consecuencia, la precisión mejora. En la (figura 5.11) podemos observar, el entrenamiento de la rede neuronal, aplicando la función de pérdida y el optimizador Adams y generando un entrenamiento por lotes de datos e iteraciones denominados épocas. Para la época 1 y cada 50 lotes podemos observar que la precisión es de 0 % de palabras correctamente predichas. Pero después de 1000 lotes, ya predecimos correctamente un 10 % de las palabras, que no son muchas pero que ya es

capaz de hacer predicciones bastante acertadas. A medida que las épocas van aumentando, la función de pérdidas va disminuyendo.

En este trabajo se llegó alcanzar el 30% de predicción correcta (ver figura 5.10), eso significa que traduce a la perfección un poco más de la cuarta parte de las frases, lo cual es un resultado más o menos aceptable, pero podría mejorar enriqueciendo los corpus.

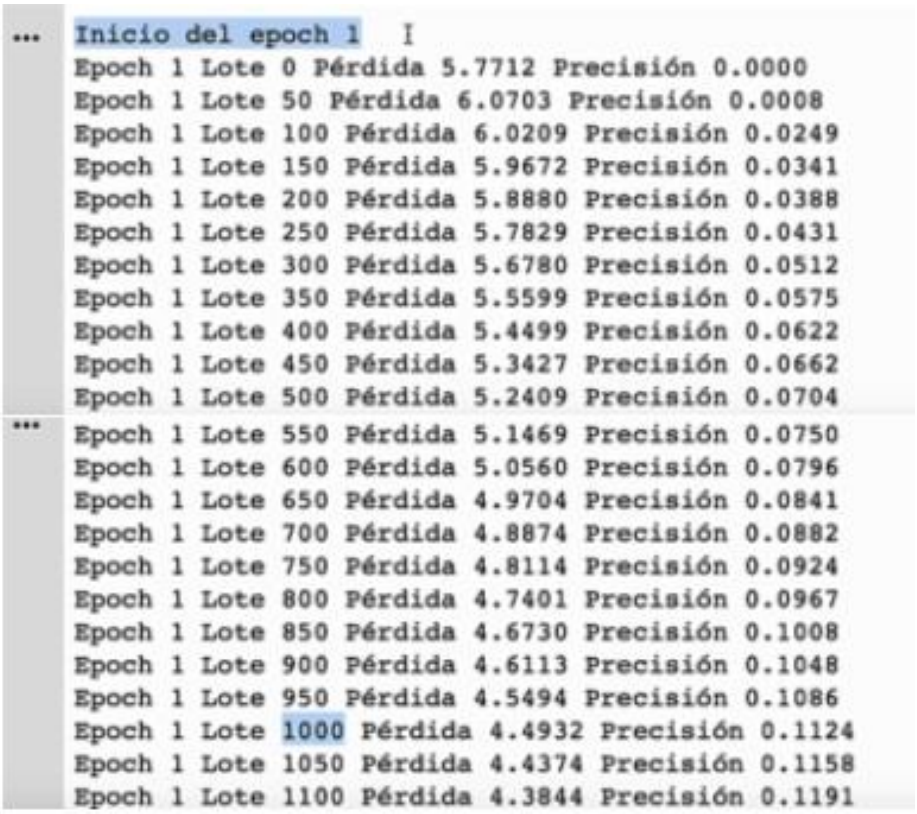


Figura 5.10: Proceso de entrenamiento de las redes neuronales.

5.2.1.3. Evaluación

En esta fase de evaluación probamos el traductor automático, esta es la parte más importante del presente trabajo, porque ya hemos construido y entrenado un traductor automático, pero se necesita saber si funciona o no ya que el hecho de que hemos obtenido buena precisión en la fase de entrenamiento no significa que el traductor

automático funciona bien. Para ello se ha creado, como se puede apreciar en la figura 5.11, una función que evalúa la entrada del modelo, la entrada es una frase, palabras que van a entrar en formato *string*, pasamos primero por un proceso de tokenización para realizar el proceso de traducción, lo que se hizo es en primer lugar tokenizar y luego para cada una de las palabras proceder a realizar la predicción.

```
def evaluate(inp_sentence):
    inp_sentence = \
        [VOCAB_SIZE_EN-2] + tokenizer_en.encode(inp_sentence) + [VOCAB_SIZE_EN-1]
    enc_input = tf.expand_dims(inp_sentence, axis=0)

    output = tf.expand_dims([VOCAB_SIZE_ES-2], axis=0)

    for _ in range(MAX_LENGTH):
        predictions = transformer(enc_input, output, False)

        prediction = predictions[:, -1:, :]

        predicted_id = tf.cast(tf.argmax(prediction, axis=-1), tf.int32)

        if predicted_id == VOCAB_SIZE_ES-1:
            return tf.squeeze(output, axis=0)

        output = tf.concat([output, predicted_id], axis=-1)

    return tf.squeeze(output, axis=0)
```

Figura 5.11: Función *evaluate* que traduce frases de un idioma origen a tokens en un idioma destino.

Esta función traduce frases del idioma origen a tokens en el idioma destino, las frases de entrada en formato *string* son tokenizadas, haciendo una traducción de los tokens de un idioma con los tokens de otro idioma, la frase de salida o destino aún no se puede leer ya que aún es una array enorme de números que simbolizan los tokens del idioma destino. Para ello, se implementó un último método que se encarga de hacer el proceso completo de traducción, el cual es el método *Translate* que se explica en la siguiente iteración.

5.2.2. ITERACIÓN 2

Una vez completada la primera iteración, en la que el motor de traducción automática ya está operativo, esta fase se centra en la traducción, desglosada del siguiente modo:

5.2.2.1. Pre edición del texto original.

Los errores ortográficos en el texto original dificultan la traducción automática. Cuando leemos en nuestra propia lengua, tendemos a corregir mentalmente los errores para comprender el sentido. El traductor automático, en cambio, al encontrar una palabra mal escrita no sabe cómo interpretarla y es muy probable que la traduzca mal o incluso produzca una frase sin sentido.

Los guiones entre palabras, símbolos poco frecuentes y signos de puntuación también afectan a la traducción. La traducción automática puede cambiar totalmente por unas comillas o por un punto mal puesto.

Para garantizar una buena calidad, se debe realizar una pre edición del texto antes de traducirla. En esta fase se corrige cualquier error y se minimizan los signos innecesarios que podrían entorpecer la tarea del traductor automático.

5.2.2.2. Traducción.

En el capítulo del estado del arte, se explicó el mecanismo de atención aplicado a un traductor automático, el cual optamos por implementar en este trabajo, donde se tiene una matriz Q del idioma origen multiplicado por la matriz K traspuesto, que es la matriz del idioma destino, este producto matricial nos da una nueva matriz que nos indica como cada uno de los elementos de Q en fila se relaciona con cada uno de los elementos de K transpuesta.

En la figura 5.12, se muestra una representación visual de cómo funciona la predicción de palabras y frases para la traducción aplicando el producto matricial QK^T, observamos cómo cada una de las palabras en español se relaciona con cada una de las palabras de la frase en quechua correspondientemente. En esta matriz resultante cada cuadrado que vemos representa un valor, que es la puntuación de la similaridad entre dos palabras, la similaridad es mayor cuanto más oscuro es el cuadrado.

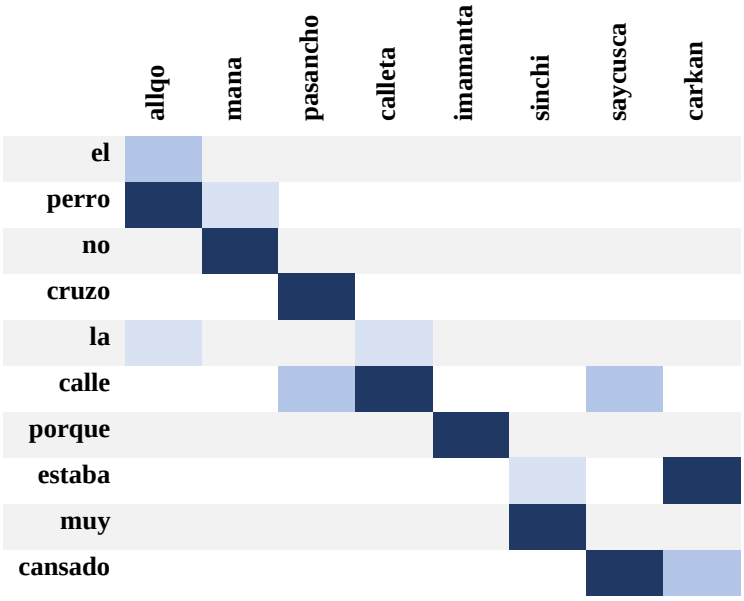


Figura 5.12: Matriz resultante del producto matricial QK^T, para obtener las similaridades más altas entre palabras del español y el quechua.

Para este trabajo, la función de predicción ingresa una oración tokenizada en el modelo y devuelve la nueva oración predicha, en nuestro caso, una traducción del quechua al español.

Estos son los pasos de ese proceso:

- Se tokeniza la oración de entrada a una secuencia de tokens.
- Se establece la secuencia de salida inicial en el *sos* token, hasta que alcanza la longitud máxima o *eos* token, y el modelo devuelve el token.
- Se obtiene el índice en el vocabulario de la palabra con mayor probabilidad.
- Se concatena la siguiente palabra predicha a la secuencia de salida.
- Y finalmente nuestra última función recibe una oración en quechua, llama al transformador para traducirla al español y muestra el resultado.

Algunos ejemplos de traducciones:

#Primera traducción:

frase = "¿Mayqintaq telefonoykiri?".

print ("Oración de entrada: {}". formato (oración))

predicted_sentence = traducir (oración)

print ("Oración de salida: {}". formato (predicted_sentence))

Oración de entrada: ¿Mayqintaq telefonoykiri?

Sentencia de salida: ¿Cuál es tu número telefónico?

 # Segunda traducción:

frase = "Ari taytay unquchkanim.".

print ("Oración de entrada: {}". formato (oración))

predicted_sentence = traducir (oración)

print ("Oración de salida: {}". formato (predicted_sentence))

Oración de entrada: Ari taytay unquchkanim.

Oración de salida: Si señor estoy enfermo.

 # Tercera traducción:

frase = "llaqtaman.".

print ("Oración de entrada: {}". formato (oración))

predicted_sentence = traducir (oración)

print ("Oración de salida: {}". formato (predicted_sentence))

Oración de entrada: llaqtaman.

Oración de salida: Mañana llegarás al pueblo.

Para este ejemplo, se experimentó con algunos valores para la dimensión del modelo y las unidades de la red de alimentación anticipada para entrenar el modelo durante una hora. Para optimizar el modelo, se debe entrenarlo durante más tiempo y con muchos valores diferentes para los hiperparámetros, así obtendremos mejores resultados.

Capítulo VI
Pruebas y Resultados

CAPÍTULO VI

PRUEBAS Y RESULTADOS

En este capítulo se muestran los resultados obtenidos de la evaluación del prototipo y se procederá a realizar una valoración en torno a los objetivos y alcances.

6.1. EVALUACIÓN

Aunque la traducción neuronal NMT es bastante reciente, ya se encuentra implementada en traductores como *Google Translate*, todavía son pocas las evaluaciones que se han realizado en cuanto a las traducciones obtenidas a través de este nuevo tipo de sistema, por lo que resulta arriesgado afirmar o desmentir que este enfoque de traducción automática es el que produce mejores resultados en bruto. Sin embargo, los primeros estudios parecen indicar que, en comparación con la traducción estadística, minimizan el esfuerzo de pos edición (Bentivogli *et al.* 2016, Castilho *et al.* 2017).

Para la evaluación del prototipo de traducción automática, se utilizó la evaluación humana bajo los criterios de fidelidad y comprensibilidad.

6.1.1. Evaluación humana

Las evaluaciones humanas permiten a los desarrolladores medir la calidad del sistema según un amplio rango de aspectos de calidad y para un conjunto de usuarios potenciales. En el capítulo anterior del marco teórico se ha propuesto un enfoque de evaluación humana, del cual nos basamos, realizando un estudio comparativo sobre los distintos niveles de la traducción automática teniendo en cuenta el parecer de los expertos humanos, considerando las siguientes variables:

- **La fidelidad o exactitud** se mide como la cantidad de información que se mantiene en la frase traducida comparada con la original, en una escala del 1 al 5, (1-ninguna, 2-poco, 3-mucho, 4-sumamente, 5-todo).

- **La comprensibilidad** indica en qué medida la traducción automática es comprensible, sin tener en cuenta el texto original, también en una escala del 1 al 5, (1-incomprensible, 2-difluente, 3-no nativo, 4-bueno, 5-impecable).

6.2. PRUEBAS DEL PROTOTIPO

Las pruebas se realizaron en una computadora con un procesador Intel(R) Core (TM) i5-2520M CPU 2.50GHz, de 64 bits, procesador x64, con 4 GB de memoria RAM y con sistema operativo Windows 10 Enterprise. El corpus fue construido de textos escritos pertenecientes a la familia quechua Cuzco-Collao que a su vez es parte del Quechua Sureño (Quechua II-C). En la tabla 4.1 se muestra las características del corpus alineado con el que se entrenó el traductor.

Para el entrenamiento de frases, codificación, decodificación y toda la implementación del código del algoritmo de *Transformer*, se usó el editor de código *Jupyter Notebook* en su versión 6.2.0, se usaron también las bibliotecas *Keras* en la versión 2.2.4-tf, dentro de *Tensorflow* en la versión 2.0.0.

	Quechua	**Español**
Frases	1.919	1.919
Palabras	12.630	22.040
Tokens	4.088	2.739
Tamaño	118 kb	124 kb

Tabla 6.1: Características del corpus usado

El tamaño del corpus es muy reducido, si se compara con el corpus Europarl (Koehn 2005) que contiene en sus idiomas más estudiados aproximadamente 2 millones de frases con alrededor de 50 millones de palabras en inglés y 44 millones de palabras en el idioma origen. Los idiomas con menor corpus contienen de 300 mil a 700 mil frases con 10 millones de palabras en inglés e igual número de palabras en el idioma origen. Por lo

cual se espera una fuerte penalización en rendimiento comparado con los sistemas que son entrenados con grandes cantidades de datos.

6.2.1. Estrategias de prueba

Las pruebas de traducción las dividimos en dos tipos, traducción de palabras y traducción de frases, para ello se necesitó la traducción y control de un experto traductor humano del idioma quechua, aplicando los criterios más usados que son, de fidelidad y comprensibilidad.

El primer caso de pruebas se realizó con palabras sueltas, llegando a resultados óptimos de traducción, con un promedio de 3.8 puntos en fidelidad y 4 puntos en comprensibilidad, como se observa en las tablas 6.2 y 6.3, donde también podemos observar que existen traducciones erróneas por lo que en criterio de fidelidad reciben un puntaje bajo debido a que en el corpus del prototipo la palabra es inexistente y se traduce a las palabras más probables.

- **Traducción de palabras:**

QUECHUA A ESPAÑOL					
No	**Texto a traducir**	**Traducido por el humano**	**Traductor automático**	**Fidelidad**	**Comprensibilidad**
1	allqo	perro	perro	5	5
2	tusuy	bailar	bailar	5	5
3	uraykuy	bajar	pequeño	1	1
4	rikchay	despertar	despertar	5	5
5	llallinakuy	competir	competir	5	5
6	tanqay	empujar	empujar	5	5
7	chiqap	bastante	cantidad	1	2
8	yana	negro	negro	5	5
9	raku	grueso	grueso	5	5
10	lumpu	esférico	suave	1	2
Promedio				3.8	4

Tabla 6.2. Resultado de la medición de la traducción automática de palabras, del quechua al español, usando la escala de evaluación de cinco puntos.

ESPAÑOL A QUECHUA					
No	**Texto a traducir**	**Traducido por el humano**	**Traductor automático**	**Fidelidad**	**Comprensibilidad**
1	sapo	hampatu	hampatu	5	5
2	cuatro	tawa	tawa	5	5
3	cabeza	uma	uma	5	5
4	amarillo	K'illo	yana	1	2
5	Pan	T'anta	yugu	1	1
6	viajar	riy	riy	5	5
7	bebè	wawa	wawa	5	5
8	pegar	waqtay	waqtay	5	5
9	pie	chaki	Chaki	5	5
10	olla	manqa	ayachukuchiy	1	2
Promedio				3.8	4

Tabla 6.3. Resultado de la medición de la traducción automática de palabras, del español al quechua, usando la escala de evaluación de cinco puntos.

- **Traducción de frases:**

  El segundo caso de pruebas se realizó con frases y oraciones compuestas, como se puede observar en las (tablas 6.4 y 6.5).

QUECHUA A ESPAÑOL					
No	**Texto a traducir**	**Traducido por el humano**	**Traductor automático**	**Fidelidad**	**Compren-sibilidad**
1	Sumaq sonqo taytayki.	Tu padre es de buen corazón.	Tu padre es de buen corazón.	5	5
2	Pay takin sapa tutan.	El canta cada noche.	El canta cada noche.	5	5
3	Manasá yachasqachu maman imayna kasqanta.	No sabía dice como era su madre.	Dice que no había madre.	1	2
4	Mariacha uwiqata michisqa.	María pasteaba sus ovejas	María pasteaba sus ovejas	5	5
5	Alqu amigunwan.	El perro y su amigo.	El perro y su amigo.	5	5
6	Asichikuq willakuyta niwaptinmi asichkani	Estoy riéndome porque me contó un chiste.	Ese rie reilón muchacho de cualquier cosa.	1	1
7	Kuchita watarqamuy wak sacha sikinpi.	Amarra al cerdo debajo de aquel	Llevaré los cerdos a la	1	2

		árbol.	colina.		
8	Chisi tuta mana puñuyta atinichu	Anoche no pude dormir	Anoche mis sueños hablando contigo	1	2
9	Atoqshi üshapa qepanta cörrir aywanaq charipänanyoq.	El zorro se fue corriendo detrás de la oveja hasta alcanzarla.	El zorro se fue a su oveja hasta alcanzarla.	2	4
10	Manuel, unayña mana tupanchikchu	Manuel, hace tiempo que no te veo	El señor Manuel es buena persona	1	5
Promedio				2.7	3.6

Tabla 6.4. Resultado de medición de la traducción automática de frases, del quechua al español, usando la escala de evaluación de cinco puntos

Para el caso de las traducciones del quechua al español, los resultados de esta segunda prueba no son tan óptimos, como se puede observar en la (tabla 6.4), ya que en el criterio de comprensibilidad llegamos a un promedio de 3.6 puntos, pero en cuanto a fidelidad, solo alcanzamos a 2.7 puntos, podemos observar que también existen traducciones erróneas por lo que en criterio de fidelidad reciben un puntaje bajo, esto debido a que en el corpus del prototipo las palabras inexistentes, se traducen a las palabras más probables que el sistema es capaz de predecir.

ESPAÑOL A QUECHUA					
No	**Texto a raducir**	**Traducido por el humano**	**Traductor automático**	**Fidelidad**	**Compren-sibilidad**
1	Anoche no pude dormir	Chisi tuta mana puñuyta atinichu	Musquyniypi chisi tuta qamwan rimachkasqani	1	2
2	¿Dónde estará mi madre?	¿Maypiraq mamay kachkan?	¿Maypiraq mamay kachkan?	5	5
3	la silla amarilla está rota	q'illu tiyanap'akiy kashan	Tiyana q'illu kashan p'akiy	1	2
4	El zorro y la oveja	Atoqwan usha.	Atoqwan usha.	5	5
5	Si siembras cosecharás	Sichus tarpunki kosechanki	Tarpuskanmi chayrayku mana hamunchu	1	2

6	El señor Manuel es buena persona	Tayta Manuel allin runam.	Manuel, unayña mana tupanchikchu	1	4
7	Alójame por esta noche.	Samachiway kunan tuta.	Samachiway kunan tuta.	5	5
8	¿Dónde están tu madre y tu padre?	¿Mamaykirí, tataykirí?	¿Mamaykirí, tataykirí?	5	5
9	busco a las gallinas	wallpakunatam maskani	Wallpañataq maskan	1	4
10	la casa blanca está bonita	Yuraq wasisumaq kashan	Wasiyuraq kashan sumaq	1	2
Promedio				2.6	3.6

Tabla 6.5. Resultado de medición de la traducción automática de frases, del español al quechua, usando la escala de evaluación de cinco puntos.

De igual forma, para el caso de las traducciones del español al quechua, los resultados de esta segunda prueba tampoco son óptimos, como podemos observar en la (tabla 6.5), ya que en el criterio de comprensibilidad llegamos a un promedio de 3.6 puntos, pero en cuanto a fidelidad, solo alcanzamos a 2.6 puntos, podemos observar que también existen traducciones erróneas por lo que en criterio de fidelidad reciben un puntaje bajo, esto también debido a que en el corpus del prototipo las palabras inexistentes se traduce a las palabras más probables que el sistema es capaz de predecir.

6.3. COMPARACIÓN DE RESULTADOS

Una vez ya teniendo los resultados obtenidos de las evaluaciones manuales, tanto para el caso de las traducciones del quechua al español y del español al quechua, se realizó la comparación en la (tabla 6.6).

Traducción de palabras	Promedio en criterio de fidelidad pts.	Promedio en criterio de comprensibilidad pts.
Quechua - español	3.8	4
	sumamente	sumamente
Español - quechua	3.8	4
	sumamente	sumamente
Traducción de frases	**Promedio en criterio de fidelidad pts.**	**Promedio en criterio de comprensibilidad pts.**
Quechua - español	2.7	3.6
	mucho	bueno
Español - quechua	2.6	3.6
	mucho	bueno

Tabla 6.6. Comparación de los resultados de medición del traductor automático del quechua al español y del español al quechua.

En los resultados podemos observar que, para la traducción de palabras del quechua al español y viceversa, se logró obtener un promedio de fidelidad de 3.8 puntos, por lo cual se considera sumamente exacta en sus traducciones y de comprensibilidad impecable con 5 puntos de obtención. Mientras que, para la traducción de frases español al quechua y viceversa, se logró obtener en fidelidad, un promedio de 2.75 puntos y en comprensibilidad 4.15 puntos, por lo cual se considera que tiene mucha fidelidad en sus traducciones y de comprensibilidad buena.

Este estudio demostró que la exactitud y la comprensibilidad tienen una alta correlación cuando los criterios de los evaluadores fueron promediados palabra a palabra y frase a frase. En las pruebas realizadas, habrá que considerar posibles sesgos por el reducido corpus utilizado para la experimentación.

Capítulo VII
Conclusiones y Recomendaciones

CAPÍTULO VII

CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

En el presente trabajo se ha implementado un traductor automático del quechua al español y del español al quechua, basado en redes neuronales (NMT). Este traductor automático se enfrenta al reto de contar únicamente con una cantidad limitada de ejemplos de traducción, es decir para esta combinación existen pocos recursos, tanto de corpus como de conocimientos gramaticales. El corpus escaso impide utilizar los sistemas NMT sin realizar modificaciones a su proceso de entrenamiento y traducción. Utilizar el modelo NMT limitaría automáticamente el traductor a un único par de idiomas, sin ampliar el caso a otros pares de idiomas nativos. La gran ventaja de utilizar NMT para la traducción, es que para idiomas semejantes no se requiere mayores modificaciones, sino únicamente un corpus alineado del par de idiomas a traducir. El texto puede ser recopilado de libros u otras fuentes o creado específicamente para el traductor automático.

El problema de traducción sigue siendo un campo propicio para la investigación y la traducción automatizada es un reto aún mayor. A pesar de encontrarse muy desarrollada la traducción automática para ciertos pares de idiomas, los resultados aún son imperfectos para otros. La cuestión se acentúa si se considera la falta de herramientas lingüísticas computacionales para idiomas como el quechua.

El presente traductor automático quechua – español, abre las puertas a futuras investigaciones para el desarrollo de un traductor óptimo que sea capaz de traducir una gran cantidad de textos, para los pueblos originarios en su propia lengua, e involucrar al hablante del español a la rica semántica del quechua.

7.2. RECOMENDACIONES

En el capítulo de resultados se muestra la capacidad de traducción que puede proporcionarse utilizando los NMT del estado del arte y se compara el corpus existente y sus resultados para estos traductores. Durante las pruebas, la implementación y la comparación, fue posible ver grandes retos, para la traducción NMT de un idioma como el quechua, para ello las siguientes recomendaciones.

- **Problema de la traducción con recursos escasos.** Como ya se ha planteado al inicio de este trabajo, los idiomas indígenas u originarios, carecen de amplias fuentes escritas que permitan su análisis. Los pocos recursos de los que se dispone deberán, por lo tanto, ser aprovechados de tal manera que se obtengan resultados aceptables.

- **Poca estandarización del lenguaje y su escritura.** Si se toma el texto en quechua, u otros idiomas indígenas, el análisis del lenguaje se enfrenta a una gran cantidad de ruido proveniente de diferentes escrituras, ortografías, conceptos de palabras, y sobre todo de dialectos dentro del mismo idioma. Si bien un normalizador y un tokenizador reducen este ruido, no logran complementar la falta de información en ciertas escrituras o las diferencias entre dialectos del mismo lenguaje. En el trabajo únicamente se trató la variante del quechua, conocida como Cuzco-Collao que a su vez es parte del Quechua Sureño (quechua II-C), ya que hay cierta comprensión mutua con el quechua boliviano.

- **Traducción de lenguas aglutinantes.** Lenguas como el quechua, toda la familia quechua en general y gran parte de las lenguas indígenas del continente americano son aglutinantes. Esto implica que, en torno a una raíz, se aglutinan morfemas que agregan significados a esta raíz. La información aglutinada de esta forma puede llegar a ser muy amplia. Las lenguas fusionantes, por el contrario,

no tienen esta capacidad y expresan menos información en sus frases. La traducción entre estas lenguas es complicada por tener topologías distintas y por contener cantidades diferentes de información por palabra, lo cual complica la alineación. Si bien, en este trabajo se ha demostrado que una alineación de frase a frase ayuda al proceso de alineamiento y traducción, no existe un equivalente entre ellos.

Referencias

Alcaraz, N. A., & Alcaraz, P. A. (2020). Aplicación web de Análisis y Traducción Automática Guaraní–Español/Español–Guaraní. *Revista Científica de la UCSA*, *7*(2), 41-69.

Álvarez García, A.; De las Heras del Dedo, R.; Lasa Gómez, C. (2011). Manual imprescindible de métodos Ágiles y Scrum. Madrid: Anaya Multimedia. (Manuales imprescindibles).

Armentano C., Corbí A., Forcada M., Ginestí M., Montava. & Ortiz M. 2007. Una Plataforma de Código Abierto Para el Desarrollo de Sistemas de Traducción Automática, Departamento de Lingüística y Sistemas Informáticos de la Universidad de Alicante.

Alzantot, M., Wang, Y., Ren, Z. y Srivastava, MB (2017). RSTensorFlow: TensorFlow habilitado para GPU para aprendizaje profundo en dispositivos Android básicos. *MobiSys ...: la ... Conferencia Internacional sobre Sistemas, Aplicaciones y Servicios Móviles. Conferencia internacional sobre sistemas, aplicaciones y servicios móviles , 2017* , 7–12. https://doi.org/10.1145/3089801.3089805

Bolivia Censo (2012): Algunas claves para entender la variable indígena. https://www.cejis.org/bolivia-censo-2012-algunas-claves-para-entender-la-variable-indigena/.

Calixto, Iacer, Stein, Daniel; Matusov, Evgeny, Lohar, Pintu, Castilho, Sheila, Way, Andy [Valencia] (2017). «Using Images to Improve Machine-Translating E-Commerce Product Listings». *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 637–643

Cavero, I. C., & Madariaga, J. F. (2007). Traductor morfológico del castellano y quechua. *Revista I+ i*, *1*(1).

Celia Rico, La formación de traductores en Traducción Automática, https://ddd.uab.cat/pub/tradumatica/tradumatica_a2017n15/tradumatica_a2017n15p75.pdf, 2017.

Corpas Pastor, G.; Seghiri Domínguez, M. (2007). "Determinación del umbral de representatividad de un corpus mediante el algoritmo N-Cor".

Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, *20*(2), 1-13.

Dunne, K. J.; Dunne E. S. (eds.) (2011). Translation and localization project management: the art of the possible. Amsterdam [etc.]: John Benjamins. (American Translators Association scholarly monograph series; 16).

Eduardo Muñoz, ingeniero de software, Towards Data Science, https://towardsdatascience.com/attention-is-all-you-need-discovering-the-transformer-paper-73e5ff5e0634, 2020.

Farid Murzone (2020). *Natural Language Processing* (NLP). Data Scientist, Developer, Sociologist and food lover.

Goldsborough, P. (2016). Un recorrido por tensorflow. *preimpresión de arXiv arXiv: 1610.01178* .

Gouadec, D. (2007). Translation as a profession. Amsterdam [etc.]: John Benjamins. (Benjamins translation library; 73).

Guillermo, B. B. (1981), Utopía y revolución: El pensamiento político contemporáneo de los indios en América Latina, 1 edn, Editorial nueva imagen, México.

Hofmann, C.; Mehnert, T. (2000). "Multilingual Information Management at Schneider Automation". En: Sprung, R. C.; Jaroniec, S. (eds.). Translating into success: cuttingedge strategies for going multilingual in a global age. Amsterdam [etc.]: John Benjamins. (Case studies in business and language).

Instituto Nacional de Estadística – INE 1 - UNFPA Bolivia (2012). https://bolivia.unfpa.org/sites/default/files/pub-pdf/Caracteristicas_de_Poblacion_2012.pdf

Jay Alammar, publicación de blog "The Ilustrated Transformer" , 2018.

Juan I. Barrios A. MD Msc. (2020). Redes Neuronales Convolucionales (CNN). Consultores estratégicos en Ciencia de Datos, BIG DATA, Ciencia de datos, Inteligencia Artificial, Aprendizaje automático.

Ketkar, N. (2017). Introduction to keras. In *Deep learning with Python* (pp. 97-111). Apress, Berkeley, CA.

Kit, C. & Wong, B. (2015). Evaluation in Machine Translation and Computer Aided Translation. En Chan Sin-Wai (Ed.). The Routledge Encyclopedia of Translation Technology (pp. 213–236). Abingdon, Nueva York: Routledge.

Laukaitis, A. & Vasilecas, O. (2007), Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, chapter Asymmetric Hybrid Machine Translation for Languages with Scarce Resources, pp. 397-408. URL: http://dx.doi.org/10.1007/978-3-540-70939-8 35

Martín-Mor, A.; Piqué, R.; Sánchez-Gijón, P. (2016): Tradumàtica: tecnologies de la traducció. Vic: Eumo. (Biblioteca de traducció i interpretació).

Moorkens, J.; Doherty, S.; Kenny, D.; O'Brien, S. (2014). "A virtuous circle: laundering translation memory data using statistical machine translation". Perspectives: Studies in translation theory and practice, v. 22, n. 3, p. 291-203. . [Consulta: 13 de octubre de 2017].

Muller, E., Bednar, JA, Diesmann, M., Gewaltig, MO, Hines, M. y Davison, AP (2015). Python en neurociencia. *Fronteras en neuroinformática* , *9* , 11. https://doi.org/10.3389/fninf.2015.00011.

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO 2019). https://archive-it.org/organizations/1424

Parra, (2014). Marie Skłodowska-Curie en Dublin City University (DCU). http://www.lalinternadeltraductor.org/n16/traduccion-automatica.html.

Peter Bloem, publicación de blog "Transformers from scratch" , 2019.

Pinnis, M.; Skadins, R.; Vasiljevs, A. (2014). "Real-world challenges in application of MT for localization: the Baltic Case". En: Beregovaya, O. [et al.] (eds.). Proceedings of the 11th Conference of the Association for Machine Translation in the Americas, Vol. 2: MT Users Track. AMTA, p. 71-84. . [Consulta: 13 de octubre de 2017].

Procesamiento del Lenguaje Natural, v. 39, p. 165-172. . [Consulta: 13 de octubre de 2017].

Saldanha, G., & O'Brien, S. (2014). Research methodologies in translation studies. Londres: Routledge.

UMSA (2018). TREINTA DE LAS TREINTA Y SEIS LENGUAS NATIVAS EN BOLIVIA ESTÁN EN PELIGRO DE EXTINCIÓN. https://www.umsa.bo/umsa-noticias/-/asset_publisher/sIpuYXdbB9M8/content/treinta-de-las-36-lenguas-nativas-en-bolivia-estan-en-peligro-de-extincion#/image/journal/article?img_id=272113&t=1554320738064

Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia, "Atención es todo lo que necesitas" , 2017.

Vilca, H. D. C. (2009). TRADUCTOR AUTOMÁTICO EN LINEA DEL ESPAÑOL A QUECHUA, BASADO EN LA PLATAFORMA LIBRE Y CÓDIGO ABIERTO APERTIUM. *Revista de Investigaciones (Puno)-Escuela de Posgrado de la UNA PUNO*, *5*(3).

DOCUMENTACIÓN

La Paz, 12 de agosto de 2021

Señor

Ph.D. José María Tapia Baltazar
Director
Carrera de Informática
Facultad de Ciencias Puras y Naturales

Presente

Ref. Aval para la defensa de Tesis de Grado

De mi mayor consideración

Por intermedio de la presente, y en mi calidad de Tutor Metodológico, tengo a bien dirigirme a su autoridad, para darle a conocer que luego de efectuar el seguimiento a la estructura y contenido de la Tesis de Grado, titulada "TRADUCTOR AUTOMÁTICO ESPAÑOL – QUECHUA, BASADO EN EL PROCESAMIENTO DEL LENGUAJE NATURAL", elaborada por el postulante René Alex Apanqui Otoya con C.I. 6952065 LP, me corresponde **dar mi CONFORMIDAD Y AVAL**, para que el mismo proceda a la **DEFENSA PÚBLICA DE LA TESIS DE GRADO**, de acuerdo a normas y reglamentos universitarios vigentes.

Sin otro particular, me despido de usted con las consideraciones más distinguidas.

Atentamente

M.Sc. Rosa Flores Morales
TUTOR METODOLÓGICO