UAI

Universidad Abierta Interamericana

Motores de traducción automática y el impacto en el rol del Traductor Profesional

Profesora de Trabajo Final: Dra. Marcela Samela

Tutoría Técnica: Ing. Claudio Milio

Alumno: Juan Milich Tear

Trabajo Final de carrera presentado para obtener el título de

Licenciatura en Gestión de Tecnología Informática

Diciembre 2021

Resumen

En el presente trabajo se investigaron distintas tecnologías y técnicas utilizadas en la traducción automática de textos. El principal objetivo fue analizar y realizar una evaluación sobre los distintos motores de traducción automática disponibles en la actualidad. Este tema se puede abordar desde diferentes aspectos. Aquí, se tomó como base el análisis de tres tipos de arquitecturas aplicadas en la traducción automática.

Con el correr de las décadas, la utilización de la tecnología en la lingüística fue mutando de rol. Principalmente, gracias al avance en la capacidad de procesamiento de los ordenadores y la aparición de Machine Learning.

Si bien el enfoque de la investigación es netamente técnico, es imprescindible analizar como se desarrolló la traducción a lo largo de las últimas décadas. De esta manera, se afianzan las bases para comprender cómo la tecnología acompañó el desarrollo en las técnicas de traducción.

Se compararon traducciones realizadas con los distintos motores y arquitecturas de traducción automática. A su vez, dichos resultados se contrastaron con traducciones generadas por profesionales del área.

Mediante el análisis de los datos y resultados obtenidos en la investigación, se pudo comprobar el impacto y participación que tiene la TA en la actualidad.

Palabras clave: inteligencia artificial, traducción automática estadística, traducción automática neuronal, machine learning

Abstract

In the present work, different technologies and techniques used in the automatic translation of texts were investigated. The main objective was to analyze and carry out an evaluation on the different machine translation engines available today. This topic can be approached from different aspects. Here, the analysis of three types of architectures applied in machine translation was taken as a basis.

Over the decades, the use of technology in linguistics has changed role. Mainly, thanks to the advance in the processing capacity of current computers and the appearance of Machine Learning.

Although the focus of the research is purely technical, it is essential to analyze how translation developed over the last decades. In this way, the foundations are established to understand how technology accompanied the development of translation techniques.

Translations performed with the different machine translation engines and architectures were compared. In turn, these results were contrasted with translations generated by professionals in the area.

By analyzing the data and results obtained in the research, it was possible to verify the impact and participation that MT has today.

Keywords: artificial intelligence, statistical machine translation, neural machine translation, machine learning

Índice

Índice de figuras

Índice de tablas

1 Capítulo 1

Introducción

1.1 Objetivo General

El objetivo general de este trabajo es evaluar las diferentes tecnologías de traducción disponibles. Comparándolas con los resultados de la traducción humana a lo largo del tiempo. Con el fin de determinar que tecnología es la mejor opción y de acuerdo con las tendencias, concluir si el rol del traductor humano puede ser suplantado en la actualidad por motores de traducción automática.

1.2 Objetivos Particulares

Como objetivos particulares del trabajo se define:

- Explicar de qué manera trabajan los motores elegidos y como se trata la traducción lingüística en la industria del software.
- Realizar una comparación de tecnologías disponibles.
- Investigar sobre el punto de vista que tiene el profesional traductor acerca de las arquitecturas de traducción automática actuales.

A partir de los objetivos anteriormente expuestos, se realiza una investigación completa acerca de las tecnologías y plataformas disponibles aplicadas a la traducción. Con el fin de lograr una conclusión sobre el impacto que tiene (desde el punto de vista tecnológico) en el Traductor profesional, paralelamente y de manera adicional, en base a las tendencias analizadas, se emite un juicio acerca de que tan necesario será estudiar un idioma extranjero en el futuro.

1.3 Justificación del tema

Hoy en día existen infinidad de aplicaciones, sistemas y programas que nos facilitan las tareas, o ayudan, a la hora de tener que traducir un texto, sin embargo, no necesitamos tener un grado académico para notar y entender que las herramientas disponibles no siempre brindan el mismo resultado. De una misma muestra de texto se pueden obtener diferentes traducciones. Además, las cuestiones culturales, regionales, incluso socioeconómicas afectan a la lingüística. Es oportuno entender, buscar una respuesta y solución con el fin de lograr descifrar estas cuestiones a través de la tecnología.

Se toma como caso de estudio el idioma inglés y un idioma de raíz eslava (croata), ambos lenguajes tienen una estructura y raíz completamente distinta, que sirve para entender tecnológicamente cual es la mejor manera de traducirlos automáticamente.
En la actualidad y con el desarrollo continuo de Machine Learning, se abre un abanico casi infinito de posibilidades a través del autoaprendizaje. Utilizando patrones y modelos matemáticos, se logra entrenar a los motores para que puedan traducir en base al contexto del texto trabajado.

1.4 Problemas, Soluciones e Hipótesis

A lo largo de este trabajo, se detalla la evolución de la traducción. Como posibles problemas se encuentran muchos resultados que no son fiables, debido a que la mayoría de las tecnologías traducen de manera literal, palabra por palabra y no hay manera de que se tengan en cuenta factores y conceptos culturales. Por otro lado, las herramientas de traducción disponibles de uso masivo utilizan el idioma inglés como lengua intermedia para luego traducir a la lengua objetivo.

El resultado de dicha traducción se ve influenciado por la lengua intermedia sacrificando calidad. Partiendo de esta base se plantea una problemática de la cual nacen varias preguntas, entre ellas, las que marcan la dirección de esta investigación:

- ¿Qué tipos de motores son fiables al momento de traducir textos completos?
- ¿El mismo motor funciona con la misma precisión para idiomas con estructuras totalmente distintas?
- ¿Se puede prescindir del traductor humano gracias a la tecnología?
- ¿Según las tendencias, en un futuro va a ser necesario estudiar un idioma extranjero?
- ¿Es posible entrenar motores de traducción con resultados de calidad y que no utilicen al inglés como lengua intermedia?

A partir de las preguntas definidas, se busca comprobar o refutar la hipótesis planteada en este trabajo. La misma afirma que, la traducción de textos literarios y científicos es una tarea totalmente automatizable en la actualidad, utilizando motores neuronales. Con la disponibilidad de herramientas open source se pueden crear y entrenar modelos totalmente funcionales que cumplan este objetivo.

1.5 Nuestra Propuesta

La propuesta definida de este trabajo es realizar y documentar traducciones de los idiomas planteados.

Estas traducciones se hacen utilizando los 3 tipos de motores definidos y desarrollados en el marco teórico, traducción basada en reglas, traducción estadística y traducción neuronal. En base a los resultados de estas pruebas, se verifica cómo se comportan las diferentes arquitecturas en los dos idiomas que fueron definidos como objetivos del análisis.

1.6 Contribuciones Principales

Las contribuciones principales y valor agregado de este trabajo son:

- La investigación profunda de 3 arquitecturas aplicadas a la traducción.
- El análisis de las redes neuronales artificiales y su aplicación en la lingüística.
- La definición del rol de la TA en la actualidad y la conclusión del motor óptimo para la automatización de la tarea.

- Plantear la solución tecnológica como respuesta a la problemática de contemplar el factor cultural en las traducciones actuales.
- Analizar la viabilidad de eliminar el idioma inglés como lengua intermedia en la TA.
- Reformular el rol del profesional traductor, de corrector a entrenador de motores de TA.

1.7 Estructura General del Trabajo

El trabajo de investigación está compuesto por 12 capítulos, en cada uno de ellos se desarrollan y describen los principales aspectos definidos del trabajo. Los anexos con la corroboración empírica de los análisis obtenidos. Acrónimos y Referencias.

1.7.1 Capítulos

- En el capítulo 2 **Marco Teórico** se detallan todas las características principales de la Traducción. Empezando por la historia de la misma, avanzando por la evolución de la TA, la problemática que genera la misma. Explicando el funcionamiento, operatoria, análisis y producción de cada una de las arquitecturas definidas en este trabajo de investigación

- En el capítulo 3 **Experimentación** se describe y detalla la metodología a utilizar en los experimentos de cada motor, junto con el detalle de los textos y oraciones a utilizar para llevar a cabo las pruebas. El objetivo es utilizar el mismo conjunto de textos en todos los motores para lograr obtener resultados que sean comparables.

- En el capítulo 4 **Traducción basada en reglas** se describe la utilización de esta arquitectura en los casos prácticos planteados. Para ello se usa la herramienta Apertium, detallando el paso a paso de su uso e instalación.

- En el capítulo 5 **Traducción estadística** se desarrollan los casos prácticos propuestos, basándose en los modelos estadísticos. Se detallan las problemáticas y necesidades encontradas en la traducción utilizando esta arquitectura. Para realizar las pruebas se

utilizan dos motores, se crea un motor basado en la Joshua y por otro lado se usa Microsoft Translator.

- En el capítulo 6 **Traducción neuronal** se necesita definir qué arquitectura neuronal utilizar para realizar las pruebas de TA. Se utiliza OpenNMT. El mismo es un framework de libre utilización y está compuesto por implementaciones de arquitecturas neuronales. El sistema es entrenado mediante una librería de corpus paralelo. En este tipo de arquitectura definiremos el concepto de sobreajuste y se implementa para así obtener los resultados más eficientes.

- En el capítulo 7 **Herramientas** se desarrolla una encuesta en campo, acerca de las herramientas más utilizadas actualmente en el rubro de la traducción lingüística. El fin de los resultados analizados arroja que tipo de arquitectura es la predominante actualmente en el ámbito académico y profesional. La institución donde se realiza la encuesta es la Escuela Normal Superior en Lenguas Vivas "Sofía E. Broquen de Spangenberg".

- En el capítulo 8 **Intervención humana** se describe, a partir de los resultados obtenidos en el capítulo 7, qué grado de intervención humana tiene el Traductor Profesional sobre la TA actual utilizada en el ámbito profesional. El objetivo es entender el grado de aprehensión de la comunidad traductora por sobre las nuevas tecnologías aplicadas en la TA.

- En el capítulo 9 **Análisis de datos** se realiza el análisis profundo de todos los resultados obtenidos en los capítulos 4, 5 y 6. Mediante la métrica BLEU son catalogados como traducciones de calidad o no.

- En el capítulo 10 **Conclusiones** se presenta la síntesis de la investigación, definiendo las conclusiones obtenidas luego del completo análisis de resultados. También se presentan las respuestas a las preguntas y problemáticas planteadas.

- En el capítulo 11 **Futuras líneas de investigación** se detallan las posibles líneas de investigación asociadas que no fueron consideradas en el desarrollo de este trabajo pero que ameritan ser tenidas en cuenta en futuros trabajos.

1.7.2 Acrónimos

TA Traducción Automática

TAE Traducción Automática Estadística

TAO Traducción Asistida por Ordenador

TAN Traducción Automática Neuronal

LO Lengua Origen

LM Lengua Meta

BLEU Bilingual Evaluation Understudy

LD Lengua Destino

TA Traducción Automática

HR Croata

EN Inglés

SP Español

RNA Redes Neuronales Artificiales

NLP Procesamiento de Lenguaje Natural

NLU Entendimiento de Lenguaje Natural

NLG Generación de Lenguaje Natural

2 Marco Teórico

2.1 Introducción

En esta sección, se abordan las bases teóricas de la investigación. Se enfoca en entender que es la traducción, cuál es su propósito, y como la misma avanzó con el correr de los años. Se detallan 3 tipos de arquitecturas de traducción automática, en orden cronológico desde su primera aparición.

Mediante evidencia concreta se demuestra el grado de avance de estas tecnologías en el área de la traducción. No solo se abordan los avances desde el punto de vista tecnológico, sino también, hay un acercamiento sobre el rol del profesional traductor humano y cómo evolucionó esta posición junto con los avances tecnológicos que son descritos. Se adentra en los problemas actuales de la TA basada en cada una de las arquitecturas, y se hace un profundo análisis de la arquitectura más importante, la traducción neuronal.

2.2 Traducción

La RAE, define a la traducción como “Interpretación que se le da a un texto”. Sin embargo, el concepto definido puede, y debe, ir mucho más allá. Como primer paso es imprescindible entender la diferencia entre traducción, e interpretación. Dos conceptos que, aunque están muy ligados entre sí, tienen definiciones completamente distintas.

La traducción consiste, en reproducir el mensaje de un texto escrito en un idioma distinto, mientras que la interpretación, es oral e implica transmitir un mensaje de un idioma X a un idioma Z. Con respecto a definir el alcance de la traducción, Parkinson de Saz (1984) interpreta que la “traducción no se limita a transmitir un mensaje, sino que puede llegar incluso a influir decisivamente en el desarrollo de una lengua” (p. 223), por lo tanto, el concepto de traducción es más profundo de lo que normalmente se cree.

La historia de la traducción se remonta a la época del descubrimiento de la piedra de Rosetta, este fue un enorme bloque de piedra de casi una tonelada en la cual estaba tallado el decreto que enaltece al faraón Ptolomeo V. Estas escrituras estaban redactadas en 3 idiomas distintos (jeroglífico egipcio, escritura demótica y griego).

Es el registro más antiguo que se tiene acerca de algún tipo de traducción. Lo siguieron las traducciones de las civilizaciones griegas y romanas, donde se buscó traducir gran parte de la literatura griega al latín. En la edad antigua, alrededor del siglo II DC, con la incipiente desaparición del idioma hebreo, se temía que el pueblo judío se quede sin registros bíblicos entendibles, es por eso, que empezaron a aparecer las primeras traducciones del antiguo testamento, luego lo siguieron traducciones del griego al latín (Delisle, 2003). Dos siglos más tarde una traducción de la biblia llamada la Vulgata fue la que un día iba a revolucionar la historia, ya que ésta versión fue impresa por Gutenberg en 1452.

Con el correr de los años, la traducción ha avanzado y evolucionado mucho. Pero un punto clave que marca el antes y el después, es la aparición de tecnologías, las cuales empezaron a suplir falencias y cumplir con objetivos que se creían imposibles. La traducción automática, empezó a tener su auge mucho antes de lo que se cree. Con los años empezaron a desarrollarse diferentes tecnologías y arquitecturas de TA. Desde un punto de vista cronológico, la arquitectura más antigua que vamos a utilizar en nuestra investigación es la traducción basada en reglas, luego vino la traducción estadística y finalmente la más moderna es la traducción neuronal.

- 70's Traducción basada en reglas
- 90's Traducción estadística
- 00's Traducción neuronal

Cabe aclarar que los motores de traducción son agrupados en base al direccionamiento y cantidad de lenguas con los que pueden trabajar (Viver Sorolla, 2008). Existen motores bilingües esto quiere decir que trabajan con un par específico de lenguas y son capaces de traducir en cualquier dirección. En estos sistemas la Lengua Origen y Lengua Meta pueden intercambiarse, ya que el motor soporta trabajar con los dos idiomas en cualquiera de los roles. En cambio, los sistemas llamados unidireccionales, solo tienen la capacidad de traducir en una única dirección (Dentro de este grupo se encuentran los sistemas utilizados en este trabajo). Los motores multilingües tienen la característica de traducir de una Lengua Origen a varias Lenguas Meta.

2.2.1 Traducción basada en reglas

La traducción basada en reglas surgió a principios de la década del 70. El proceso se nutre de reglas lingüísticas, analizadores sintácticos y morfológicos creados por profesionales. Todas estas reglas creadas conforman diccionarios y glosarios, conformados por millones de líneas, que se encargan de alimentar al motor de TA. Utilizando toda la información de las reglas semánticas, el motor realiza el correspondiente procesamiento para encontrar la equivalencia del texto objetivo a traducir entre la Lengua Origen y la Lengua Meta.

El objetivo es alimentar los motores con la información lingüística enriquecida, de esta manera se alcanzarán resultados de calidad. Dentro de la TA basada en reglas podemos dividir tres técnicas utilizadas.

- Traducción directa
- Traducción por transferencia
- Interlingua

2.2.1.1 Traducción directa

Esta técnica, se basa en traducir cada palabra del texto al idioma de destino sin tener en cuenta su orden. No posee ningún estudio o análisis morfológico automatizado, es el tipo de traducción automática más básica. Para ello, el algoritmo utiliza un diccionario en el cual consta de palabra origen y palabra meta. A lo sumo utiliza alguna regla de ordenamiento de acuerdo al lenguaje de origen y lenguaje meta. Un punto a detallar es que, en la técnica de TA directa, no se analiza de ninguna manera la sintaxis del texto perteneciente a la Lengua Origen, por ende, es muy normal acarrear grandes errores de traducción con esta técnica, simplemente por el hecho de traducir textos de la Lengua Origen con errores sintácticos o de vocabulario.

Con el correr del tiempo, la traducción directa fue avanzando y se le agregaron análisis morfológicos y reordenamiento, para ello lo que se planteó es, trabajar por medio de traducciones incrementales. Obviamente este tipo de cambio ya no apuntaba a traducciones palabra por palabra, sino que se toma como objetivo una oración. Incrementalmente se iba traduciendo de acuerdo a algunas reglas y análisis que se parametriza a través de diccionarios

bilingües y de esta manera se lograba tener una oración completa traducida a través de la TA directa.

2.2.1.2 Traducción por transferencia

Una estrategia para vencer las diferencias entre lenguajes es modificar la estructura del texto origen para que cumpla con las propiedades del objetivo a través de reglas lingüísticas. La técnica de transferencia tiene como premisa editar el texto de la Lengua Origen para que cumpla con los atributos que necesita la Lengua Objetivo. Esta técnica se realiza siguiendo una secuencia de tres pasos.

1. Análisis
2. Transferencia
3. Generación del contenido

Olmedo ha afirmado lo siguiente:

> A través del análisis morfológico que se realiza en la primera fase, se consigue identificar los componentes del texto original y clasificarlos según su función; seguidamente, se realiza la transferencia léxica, donde se analiza el contenido y se resuelven las ambigüedades y, por último, se lleva a cabo la generación, donde se realiza la transferencia léxica. Aquí a cada palabra del texto original se le asigna un equivalente en el idioma destino (Olmedo Ruiz, 2018, p. 26)

Si nos detenemos en el análisis morfológico, se busca que cada una de las palabras este identificada con su identidad morfológica, es decir, identificar verbos, adjetivos incluso el sujeto de una oración. El análisis sintáctico, está basado en etiquetar cada una de las secciones del texto de acuerdo a que posición ocupa en la estructura. Cuando se habla de análisis de estructura, se analiza el texto en busca de expresiones idiomáticas, ya que por ejemplo las metáforas, no pueden ser traducidas de manera directa. Al analizar la Lengua Origen en fracciones mucho más amplias, es posible detectar casos donde se requiera un análisis y posterior traducción diferenciada.

La etapa de transferencia lo que hace es intentar eliminar las diferencias lingüísticas entre la Lengua Origen y la Lengua Meta. Esta transferencia se realiza a través de reglas, con

diccionarios o glosarios. La última etapa llamada Generación, se encarga de crear la traducción aplicando reordenaciones en el caso de que sea necesario. Si bien esta técnica se considera superior a la TD, los resultados están directamente relacionados con la calidad y cantidad de reglas escritas y previamente cargadas por profesionales lingüísticos. Esta técnica fue utilizada mayormente en sistemas bilingües y unidireccionales ya que resulta altamente costoso desarrollar reglas para un motor multilingüe de estas características.

2.2.1.3 Interlingua

Esta técnica se basa en generar y representar el texto, de la Lengua Origen y Lengua Meta, en una lengua intermedia, llamada interlingua. Se analizan de manera semántica las oraciones de la lengua origen y se intenta representarlo en el denominado interlingua. A partir de este lenguaje intermedio se puede generar la traducción al lenguaje objetivo, ya que se encuentran cargadas las reglas capaces de interpretar y traducir de interlingua a la Lengua Meta.
El objetivo que tiene esta técnica es que una vez que una oración está representada en interlingua, entonces podría ser traducida a cualquier Lengua Meta. Originalmente se creó con el concepto de que interlingua sea lo suficientemente potente para servir de lengua intermedia en la traducción de cualquier idioma.

Interlingua busca representar con conjunto de símbolos el significado o cierto significado de las palabras. El sistema interlingua está basado en dos módulos, el primero para el análisis y codificación a este lenguaje intermedio, y el segundo módulo es el encargado de la generación del texto desde interlingua a la Lengua Meta.

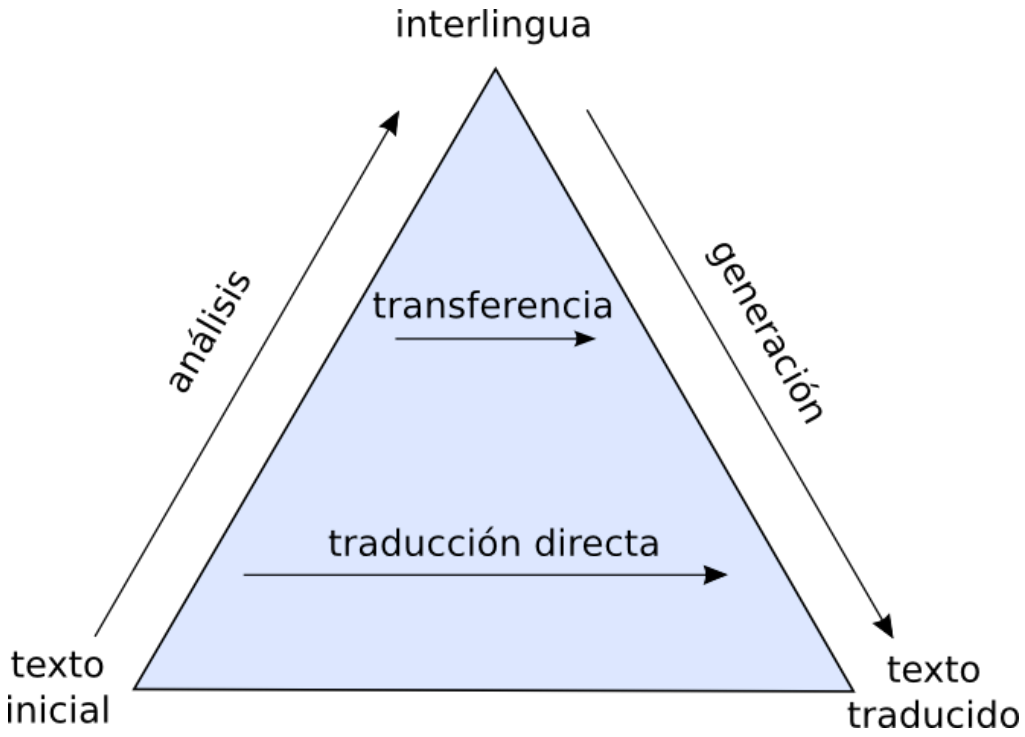

Figura 1 – Pirámide de traducción

Uno de los inconvenientes que presenta esta técnica se da cuando una palabra tiene más de dos formas posibles de traducción en la Lengua Meta. Un ejemplo claro es la palabra ‘Cielo’, en ingles puede ser interpretada como Sky o Heaven y ambas tienen connotaciones distintas.

2.2.2 Traducción estadística

A mediados del siglo XX, se empezaron a plantear conceptos acerca de las tendencias tecnológicas en cuanto a la traducción. Se empezó a considerar que por medio de la computación se iba a tener la capacidad de acceder a traducciones a través de modelos estadísticos. En el año 1949, Warren Weaver introduce el concepto de utilizar métodos estadísticos para explotar la traducción automática, pero debido a la potencia de procesamiento que tenían las computadoras en esa época, no existía la posibilidad de experimentar sobre los planteos que básicamente eran todos hipotéticos.

Hoy en día existen cientos de modelos usados en TAE, y se basan en la utilización de técnicas estadísticas para construir traducciones. Esta metodología y arquitectura de traducción necesita ser entrenada, y se logra a través del denominado corpus paralelo.
El corpus paralelo se forma mediante dos textos alineados, uno en su lenguaje original y el otro traducido con el lenguaje meta. Esta alineación va creando uniones entre las palabras de un texto y el otro. Se construyen las asociaciones entre las traducciones y de esta manera, a través de modelos estadísticos y probabilísticos, se identifican las correspondencias, tanto en palabras, como frases completas.

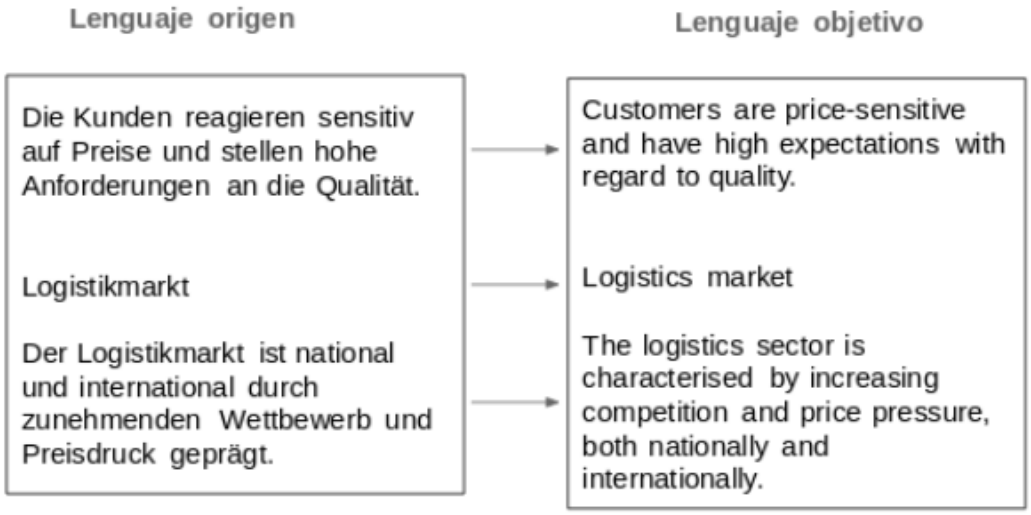

Figura 2 – Corpus Alineado

Para explicar el concepto de corpus paralelo, Weaver los hizo a través de la siguiente analogía:

> Piensa en los individuos que viven en una serie de altas torres cerradas, todas erigidas sobre una base común. Cuando tratan de comunicarse entre sí, gritan de un lado a otro, cada uno desde su propia torre cerrada. Es difícil hacer que el sonido penetre incluso en las torres más cercanas y la comunicación es muy deficiente. Pero, cuando un individuo baja de su torre, se encuentra en un gran sótano abierto, común a todas las torres. Aquí se establece una comunicación fácil y útil con las personas que también han descendido de sus torres.

> Por lo tanto, puede ser cierto que la manera de traducir del chino al árabe, o del ruso al portugués, no es intentar la ruta directa, gritando de torre en torre. Tal vez el camino sea descender desde cada idioma, hasta la base de la comunicación humana –el lenguaje universal real pero aún no descubierto- y luego resurgir por lo que sea conveniente (Weaver, citado en Hutchins, 1999, p. 6).

El uso del corpus paralelo es clave en permitir ir más allá del análisis sobre el de la traducción en sí mismo, es decir buscar la equivalencia entre el texto original y lo que se traduce, sino que mediante el corpus se puede pasar a la "investigación de la caracterización de la lengua en las traducciones" (Hallebeek, 1997, p. 7). Esto quiere decir que es posible obtener características para traducir, como las denominadas estructuras gramaticales. La TAE tiene como objetivo analizar e identificar patrones que se vuelvan recurrentes y a su vez, a través de la estadística, éstos son elegidos y designados. Cuando un motor estadístico analiza un texto, probablemente encuentre varias traducciones posibles, mediante el análisis de los corpus, determina cuál es la correcta y de esta manera, estadísticamente, llegar a resultados de calidad.

Los sistemas de TAE trabajan bajo el concepto de que al momento de traducir cualquier oración o texto de la lengua origen a la lengua objetivo, todas las oraciones de la lengua meta son una potencial traducción correcta. Pero, algunas oraciones van a tener una posibilidad mayor de ser "la traducción correcta". Con la búsqueda de patrones en los corpus, es como la TAE puede realizar los cálculos probabilísticos correspondientes para determinar qué oración es la traducción acertada. El primer modelo estadístico que se desarrolló fue el modelo basado en palabras. En este modelo la traducción se realiza palabra por palabra de acuerdo con cómo está distribuida en el corpus. Por ende, aquí, no se tiene en cuenta el contexto y por esta razón es muy difícil llegar a un resultado de traducción de calidad. Debido a que una palabra en la lengua origen puede tener múltiples traducciones en la lengua destino, surgió la necesidad de desarrollar un modelo estadístico de traducción por frases. En este caso la unidad de traducción deja de ser la palabra para pasar a ser un conjunto de las mismas.

Las frases elegidas como traducción son seleccionadas netamente bajo un modelo estadístico y sin ningún tipo de significancia lingüística. Por eso la traducción estadística introduce un concepto más, que se llama modelo de lengua.

El modelo de lengua tiene la función de que, una vez hecha la traducción estadística, se aplica nuevamente un modelo probabilístico para constatar que la traducción sea correcta y fluida en la lengua de destino. “Un modelo de lengua es un modelo que expresa cómo de verosímil es una oración en el idioma de destino. Dicho modelo se basa en n-gramas, es decir, en segmentos más pequeños que la oración. El producto final se obtiene de forma automática mediante grandes cantidades de textos en la lengua destino” (Olmedo Ruiz, 2018, p. 36).

Se puede resumir que los motores de TAE utilizan dos modelos, el de traducción, encargado de detectar los patrones y frecuencias en los corpus. Mediante estos patrones se encuentra la traducción correcta basándose en los grados de probabilidad determinados por el entrenamiento. Por otro lado, se encuentra el modelo de lengua, el cual se encarga de determinar y valorar la calidad de la traducción. La TAE utiliza un corpus extra para ajustar los parámetros y optimizar su modelo. El nombre que se le da a este corpus set de validación o set de tuning. Por lo general al corpus recolectado se lo particiona para utilizarlo en las etapas de entrenamiento, validación y testeo. Cada motor estadístico utiliza algoritmos de optimización que pueden estar catalogados en dos grupos, algoritmos batch u algoritmos online. Los algoritmos de tipo batch trabajan por medio de iteraciones del proceso completo de entrenamiento, mientras que los algoritmos online, realizan la fase de tuning cada vez que una frase o palabra es decodificada.

2.2.2.1 Arquitectura

La arquitectura de traducción estadística se basa en dos procesos principales y separados.

- El entrenamiento
- La decodificación

El entrenamiento consiste en lo descrito al principio de este capítulo, analizar y detectar patrones recurrentes sobre los corpus utilizados para realizar el entrenamiento en sí.
El proceso de decodificación es el que genera la traducción propiamente dicha. Consiste en analizar el texto a traducir y buscar estadísticamente de todas las traducciones posibles cuál es la adecuada. Si bien el grado de inferencia humana en la TAE es importante, debido a que es necesario la creación del corpus y su correcta alineación, no tiene punto de comparación con la traducción basada en reglas, donde se necesita que un profesional defina toda la serie de reglas en la creación de los glosarios.

2.2.3 Traducción neuronal

La traducción automática neuronal, es una evolución de la TAE, pero como su nombre lo indica, utiliza redes neuronales artificiales.

Estas redes son definidas como “redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico”. (Galán y Martínez, 2007, p. 3)

Las redes neuronales tienen la capacidad de aprender de la misma manera que aprende un sistema biológico. Para concertar el aprendizaje, es necesario que se le brinde información de entrada e información de salida. Para el caso de la traducción, se utilizan los corpus de entrenamiento donde se brinda en la entrada la lengua origen y en la salida la lengua meta. Las neuronas se encuentran organizadas en diferentes conjuntos que se denominan capas. A su vez la red neuronal se divide en 3 capas, capa de entrada, capa oculta y capa de salida.

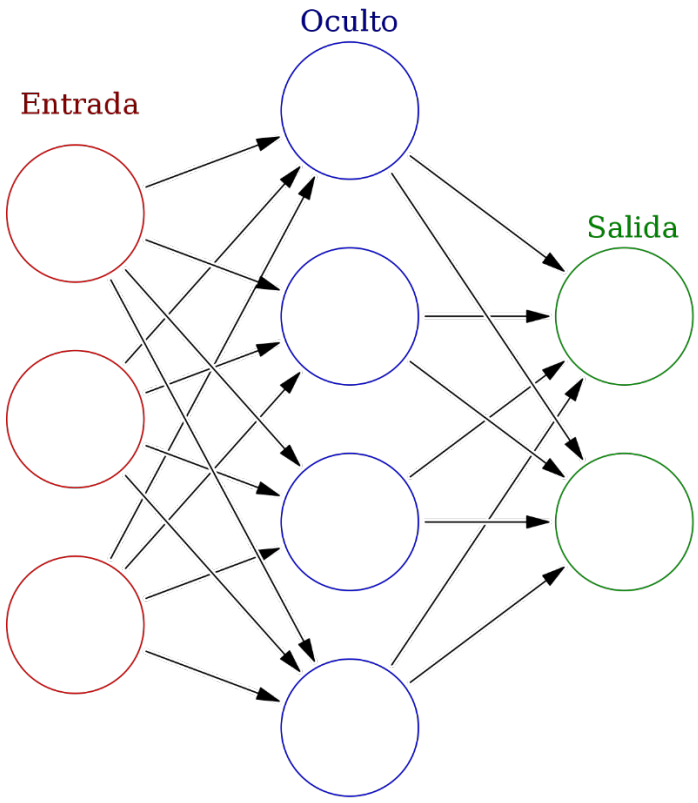

Figura 3 – Diagrama de Red Neuronal

En este esquema se detalla la estructura típica y básica de una red neuronal artificial. En la capa de entrada se encuentra el texto origen. Al ingresar dentro de la red neuronal, el texto se divide en secciones formando datos individuales. Dentro de la capa oculta, estos datos son procesados y analizados. Aquí dentro se realizan las traducciones, generando miles de traducciones posibles. El trabajo de la red neuronal es descartar todas las variables hasta dejar una sola que considere correcta y entregarla en la capa de salida, la selección de la traducción correcta es capaz hacerla en base al entrenamiento previo que tuvo la red neuronal.
Cabe detallar, que la capa oculta no es una sola, sino que, pueden ser varias capas trabajando en conjunto. Los tipos de neuronas que trabajan en varias capas se denominan perceptrones multicapa. “Los modelos de traducción automática neuronal requieren sólo una fracción de la memoria que necesitan los modelos tradicionales de traducción automática estadística”. (Merriënboer, Bahdanau, Bougares, Schwenk y Bengio, 2014, p. 103). Existen distintos tipos de redes neuronales aplicadas a la TA. Entre ellas se encuentran las redes neuronales convolucionales (CNN).

Según Wulliamoz (2018) en la CNN, el intercambio de información es llevado a cabo de manera jerárquica. Es decir, cada neurona toma una señal de entrada y aplica un filtro sobre ella. La CNN tiene muchos procesos por dentro, y la velocidad computacional es un factor clave en la performance. Este tipo de redes se utilizan mucho en el procesamiento de visión artificial. Otro tipo de redes neuronales son las denominadas redes neuronales recurrentes (RNN). Las redes recurrentes se denominan de esta manera porque procesan datos que tienen dependencia en datos anteriores, es decir, las neuronas cuentan con memoria y tienen en cuenta procesamientos anteriores. De esta manera las neuronas logran retroalimentarse con sus propias salidas. Para llegar a esto las neuronas utilizan el estado oculto. El estado oculto es la memoria de trabajo de la red, la misma contiene información del dato actual procesado y se utiliza para procesar el nuevo dato. Estos tipos de redes neuronales son utilizados en el procesamiento del lenguaje natural donde al momento de utilizarlo en la traducción, puede tener en cuenta el contexto. Donde se tiene la oportunidad de aprender particularidades de cada idioma.

Con el desarrollo de las redes RNN se introduce el concepto de Deep Learning. “Deep learning consiste en obtener un acercamiento más profundo y preciso al procesamiento cerebral humano”. (Pierre y Arteaga, 2015).
La profundidad, en el término Deep Learning, está relacionada a la cantidad de capas que se usan en la red neuronal, esto quiere decir que cuantas más capas se hallan en la red, más distancia hay entre la entrada y la salida.

El aprendizaje llevado a cabo por las redes puede ser ejecutado de dos maneras, supervisado o no supervisado. El aprendizaje se construye a partir de los datos que se utilizan en esta fase de entrenamiento. En el aprendizaje supervisado se encuentra un ente externo que inspecciona el entrenamiento, comparando el resultado deseado y el que se obtiene por medio de la red neuronal. Con esto queremos decir, que como previamente, se conoce el resultado deseado, entonces, el mismo es utilizado para guiar y ajustar todo el entrenamiento.
En cambio, en el aprendizaje no supervisado, no se conoce previamente el resultado, sino que este es descubierto por el propio proceso de aprendizaje. Lo que se busca con este tipo de aprendizaje, es que la red misma aprenda con los datos, que la red misma pueda tomar las decisiones y tenga la capacidad de llegar a los resultados, sin una acción humana de por medio.

Como se detalla anteriormente, al utilizar redes neuronales en la traducción automática, se usan redes de tipo RNN, con una arquitectura de tipo codificador - decodificador. En esta arquitectura se utiliza un codificador y decodificador para cada lengua. En la TAN, el texto se representa en vectores, es así como el codificador tiene una oración como input y lo que hace es convertirla en un vector.

La oración de input se representa de la siguiente manera:

$$x = (x_1, \ldots, x_{T_x})$$

El codificador, transforma el input en un vector utilizando las siguiente funciones

$$h_t = f(x_t, h_{t-1}) \qquad c = q(h_1, \ldots, h_{T_x})$$

La función h_t es el estado oculto en la instancia de tiempo t. La función c es el vector que se obtuvo con el procesamiento de la secuencia de estados ocultos. Las funciones q y f dependen del tipo de arquitectura RNN codificador - decodificador que se utilice. En este punto, por lo general se utilizan funciones exponenciales normalizadas.
El decodificador se encarga de predecir las palabras en base al contexto brindado por el vector c y por todas las palabras que fueron precedidas anteriormente.

2.2.3.1 Lenguaje natural

Al hablar de inteligencia artificial aplicada a la lingüística, es necesario nombrar el Procesamiento de Lenguaje Natural (NLP). Éste se enfoca en desarrollar y mejorar técnicas y sistemas abocados a optimizar la comunicación entre las personas y los ordenadores.
En ejemplos como chatbots o asistentes virtuales, se puede ver el desarrollo y el rol que cumple el NLP en la cotidianeidad. Si bien el NLP es un campo de la ciencia de la computación complejo, y en constante progreso, se divide en tres procesos con propósitos definidos:

NLP: Entiende y encuentra el significado de lo que el humano dice o escribe, y a partir de eso se define la acción a realizar generando una respuesta al humano en el mismo lenguaje original.

NLU: Comprensión para analizar datos no estructurados y habilidad para convertirlos en data estructurada.

NLG: Proceso que se encarga de producir lenguaje natural mediante fuentes de datos estructurados.

Tal como constata Lloret, Elena y Suárez, Armando (2021), el NLP, con todos sus procesos, es una tarea compleja, en constante crecimiento, y que requiere principalmente, de mucho gran investigador.

2.2.3.2 Sesgo y varianza

En la matemática estadística, se busca recolectar conclusiones de poblaciones totales utilizando muestras específicas. Cómo es realmente costoso utilizar la población total, se utilizan muestras para analizar, en las cuales los resultados son tenidos en cuenta como representativos de la población general. Existen varios factores que pueden hacer que las conclusiones extraídas sean diferentes a los datos reales. Existen particularmente dos errores cuantificables que nacen a partir de las desviaciones en las estimaciones obtenidas. Estos son el sesgo y la varianza. En resumen, el sesgo determina qué tan lejos se encuentra el valor estimado o predicho, del valor real. La varianza, en cambio, detalla la diferencia que existe entre distintas muestras que pertenecen a la misma población. (Padilla, 2012).

Estos dos conceptos estadísticos se extrapolan al aprendizaje automático realizado por las redes neuronales artificiales. En el caso del sesgo, está indirectamente relacionado con los

modelos que se utilizan para realizar las predicciones. Cuanto más complejo sea el modelo utilizado, menos probabilidad de sesgo se obtiene. La elección del modelo a utilizar en el aprendizaje automático se basa en la tarea a realizar, no se utiliza ni el mismo modelo, ni la misma arquitectura para entrenar un motor que realice traducciones o entrenar un motor que realice tareas de clasificación. Con respecto a la varianza, este concepto se relaciona, para nuestro caso de estudio, con los corpus que se utilizan tanto para entrenar el modelo, y para realizar la validación.

En este punto ya entendemos que el modelo no memoriza valores, sino que encuentra y reconoce patrones. Es por eso que es común encontrar que en la fase de entrenamiento tengan un resultado positivo, pero en la validación se encuentran fallos y predicciones erróneas. A partir de este punto es donde entra el concepto de ajuste, y las técnicas utilizadas para reducir tanto la varianza como el sesgo.

2.2.3.3 Ajuste

Las principales razones por las cuales un modelo neuronal brinda resultados pobres o erróneos. Son los referidos al ajuste, es muy común encontrar inconvenientes de sobreajuste o infra ajuste. Estos errores, particularmente, hacen referencia a los inconvenientes de generalización que pueden tener los modelos. Cuando los datos de entrenamiento que se utilizan son muy escasos, entonces se obtiene un problema de infra ajuste o underfitting. Por ejemplo, si se está entrenando un modelo para la clasificación de flores mediante imágenes, y en los datos de entrenamiento solo utilizamos una imagen de una rosa roja, entonces el modelo solo será capaz de reconocer como flor a la rosa roja, cuando se encuentre con una imagen de un clavel, no será capaz de clasificarlo como flor.

En cambio, si en la fase de entrenamiento utilizamos 100 imágenes de flores distintas, pero todas son de color rojo. Entonces, cuando el modelo se encuentre con una rosa blanca, no la reconocerá como flor tampoco. Debido a que aprendió que una condición obligatoria para que un elemento sea clasificado como flor, debe ser de color rojo. Este es un caso típico de sobreajuste u overfitting.

Tanto el sobreajuste como el infra ajuste, son problemas donde, mediante el entrenamiento no se permite que el modelo generalice el conocimiento, y por ende no es capaz de brindar predicciones de calidad. Para reducir tanto el sobreajuste como el infra ajuste se aplican diferentes técnicas. Pero como base, siempre se parte del set de datos que se

utiliza en el entrenamiento. Por eso es importante encontrar el punto medio donde se encuentre un balance y equilibrio en la fase de aprendizaje. Con el set de datos o corpus de entrada, se realiza una subdivisión del mismo para utilizar parte de la muestra en la fase de entrenamiento, y la otra parte para la fase de test o validación. Se suele utilizar el 80% del corpus para entrenar y el 20% restante para validar.

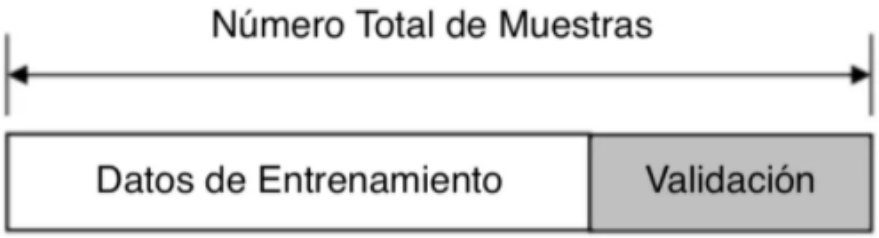

Figura 4 – Corpus Entrenamiento/Validación

Es importante que el set de datos utilizado en la validación, cuente con muestras diversas para realizar una comprobación lo más ajustada posible.

2.2.3.4 Hiperparámetros

Cuando se realiza el entrenamiento del modelo, hay hiperparámetros que se utilizan para definir la estrategia a utilizar por el algoritmo. Las redes neuronales usan numerosos hiperparámetros. Entre ellos se encuentran las funciones de activación, el parámetro de aprendizaje (learning rate), cantidad de capas, etc. No existe una estrategia única al momento de la elección de los hiperparámetros, sino que, estos permiten ajustar el comportamiento del modelo, y se eligen en base a varios factores tales como, el objetivo que tiene el modelo, la capacidad computacional y la disponibilidad y tipo de datos que se utilizan para el entrenamiento.

"No existe una teoría formal que permita acotar de manera eficiente una selección adecuada de hiperparámetros, para cualquier tarea a realizar". (Velo Fuentes, 2020).
La elección de los mismos se hace en base a experiencias o estudios previos, a la práctica o también se realiza a partir de métodos automatizados, donde se definen rangos de valores posibles para cada uno de ellos y se monitorea como rinde el modelo en base a las

combinaciones de todos los hiperparámetros. En la práctica de un escenario real, una red neuronal contiene múltiples hiperparámetros disponibles por ende resultaría imposible analizar todas las combinaciones de los valores para cada uno de los hiperparámetros. Por ejemplo, si se tienen 9 hiperparámetros con 8 valores por cada uno, se debe realizar el entrenamiento 9^8 veces sobre el set de datos de validación. La capacidad computacional es esencial en estos escenarios, para poder llevar a cabo todas las iteraciones de entrenamiento necesarias.

Una de las técnicas más utilizadas es realizar varias iteraciones e ir reduciendo el rango escogido para los valores de los hiperparámetros que ofrezcan un mejor resultado. Para comparar los resultados obtenidos en base a cada configuración de hiperparámetros, se utiliza el método de validación cruzada. Validación cruzada es un método estadístico que permite evaluar algoritmos de aprendizaje basándose en su desempeño. Los principales hiperparámetros disponibles en una red neuronal para su configuración son los siguientes:

- HiddenLayerSize: Es el número de neuronas existentes dentro de la capa oculta.
- LearningRate: Es el desplazamiento de ajuste de los pesos en cada ejecución de aprendizaje.
- EpochNumber: Cantidad de ciclos en que los datos pasarán por la red neuronal. Si se definen 4 ciclos y se tienen 100 datos, entonces, en cada ciclo, los 100 datos serán procesados por la red neuronal.
- ActivateFunction: Aquí se define cual es la función de activación a utilizar en cada neurona. Esta función es la encargada de definir cuando una neurona se activa o inactiva.
- BatchSize: Cantidad de ejecuciones internas que tiene cada epoch (ciclo). Cada ejecución se denomina iteración
- NoiseScale: Es la escala del ruido que genera la neurona.
- Tolerance: Tolerancia definida, para la detección de que la red neuronal ha convergido y no es necesario seguir entrenando. Se utiliza en la técnica de early stopping.

Regularización

Para mitigar el sobreajuste, existen varias técnicas que se utilizan con este fin. Estos métodos son llamados técnicas de regularización, y la elección de cual usar está directamente relacionado con la arquitectura del modelo. Las técnicas más comúnmente utilizadas son:

- DropOut: Esta técnica se basa en desactivar aleatoriamente neuronas, reduciendo de esta manera la complejidad de la red neuronal. La intensidad del DropOut (la probabilidad de que la neurona se active o inactive) se define como hiperparámetro.
- Early Stopping: Esta técnica se basa en detener el ajuste en el punto donde se busca reducir el error de validación cruzada. Aquí se aplican reglas para entender cuándo es el momento justo donde es necesario para el entrenamiento. Para ello se monitorea el rendimiento y resultado en cada uno de los epoch.
- Data Augmentation: En esta técnica se aplican transformaciones en los inputs de datos originales, y así se obtienen datos de entradas diferentes pero iguales en esencia, enriqueciendo la data de entrada. Esta técnica se utiliza mucho en el procesamiento de imágenes.

2.3 Trabajos relacionados

Es importante detallar como marco teórico los trabajos relacionados en el área. Para describir el escenario actual de trabajos en este campo, se detallan las últimas publicaciones de la IAMT (International Association for Machine Translation)

2.3.1 Evaluating the usefulness of neural machine translation for the Polish translators in the European Commission

En este trabajo (Karolina Stefaniak, 2020), analiza en detalle, el impacto que genera la implementación de traductores neuronales en la Dirección General de Traduccion Europa (DGT) donde se emplean alrededor de 2000 traductores de todas las lenguas oficiales europeas, con el finde ayudar a las comisiones europeas a comunicarse con los habitantes de la UE. En los últimos años los pedidos de traducción crecen, mientras las contrataciones de traductores humanos bajan. Basándose en el par de lenguas Polaco > Inglés, se comparan traducciones automáticas neuronales con las hechas por los profesionales con el fin de

determinar la rigurosidad, y la posibilidad de aumentar el flujo de producción sin sacrificar calidad en los resultados.

2.3.2 What's the Difference Between Professional Human and Machine Translation? A Blind Multi-language Study on Domain-specific MT

Aquí, (Fisher y Laubli, 2020) a partir de observar como la traducción automática genera errores que son necesarios solucionar en una fase de posedición de la misma manera que se utiliza esta fase para encontrar errores en traducciones humanas. Se realiza el experimento de presentar varias traducciones a correctores especializados, sin indicar cual traducción fue hecha de manera automática y cual se realizó a través de un traductor humano. Al final del experimento se expone y detalla donde se generan los principales inconvenientes de la TA y que diferencia se encuentran con errores humanos.

2.3.3 Document-level Neural MT: A Systematic Comparison

(Lopes, Farajian, Bowden, Zhang y Martins 2020) Proponen una discusión teórica comparando las soluciones actuales de TAN. Partiendo desde la recientemente propuesta ariquitectura "Star Transformer". Se detalla la necesidad de empezar a utilizar no solo la métrica BLEU para realizar comparaciones entre motores, sino que es necesario tener en cuenta variables muy importantes como la coherencia y la cohesión.

3 Experimentación

3.1 Introducción

En este capítulo, se presenta y detalla la experimentación llevada a cabo para cada uno de los sistemas de TA. Se procede a explicar la metodología a aplicar aplicado en cada una de las arquitecturas, la selección de los sistemas y la recolección de Corpus en los casos que esto aplique.

3.2 Metodología

Para llevar a cabo el experimento, análisis y comparación se seleccionó un set de datos conformado por frases, oraciones y fragmentos cortos de textos. Como afirma Koehn (2007) la evaluación de las traducciones de oraciones largas puede resultar muy complicada, puesto que los sistemas de TA generan traducciones muy confusas y con errores en diferentes partes. Si bien Koehn afirmó lo citado en el año 2007, la TA avanzó enormemente con el impulso de los modelos neuronales, es por eso que en el set de datos también se agregaron expresiones idiomáticas y perífrasis verbales, dos escenarios que siempre fueron problemáticos en los sistemas basados en reglas. El objetivo de esta investigación experimental es evaluar cómo responde el motor neuronal frente a estas dificultades lingüísticas. Al momento de realizar el análisis de datos obtenidos se va a tener en cuenta que con Apertium difícilmente se obtiene un resultado de calidad en estos dos escenarios específicos.

3.3 Elementos a analizar

Para cada par de lenguaje se seleccionaron los elementos a analizar. Cada uno de los elementos se encuentran en su LO y LD equivalente, dicha traducción fue hecha por un profesional del rubro. A continuación, se presentan los elementos que se analizan junto con su traducción correspondiente, luego de la experimentación se comparan y analizan los resultados obtenidos contra las traducciones humanas y así obtener las métricas correspondientes.

3.3.1 Croata > inglés

3.3.1.1 Oraciones y Frases

Tabla 1

Listado de oraciones a traducir hr>en

Segmento original	Lengua meta
Stanuje blizu škole.	He/She lives near the school.
To je između naš.	This is between us. / This stays between us.
Od nedjelje nema kiše.	It won't rain on Sunday.
Prošao je mimo svoje stare kuće.	He passed/go through his old house.
Tramvaj ne vozi zbog nestanke struje	The streetcar is not working due to power shortage.
Kuća je pored škole	The house is next to the school.
Radnici su dobili plače unatoč krizi	The workers got their salaries in spite of the crisis.

Tabla 2

Listado de frases a traducir hr>en

Segmento original	Lengua meta
Ja pjevam sada.	I'm singing.
Stalno čitam.	I keep reading.
Čitam novine	I read the newspaper.
Pada li kiša?	Is it raining?
Ta knjiga je za mene.	That book is for me.
Pica leti nada kucu.	The bird flies over the house.
Osjećam da mi fali zraka..	I feel like I can't breath. / I feel like I'm short of breath.

3.3.1.2 Texto

El texto que se utiliza como muestra para analizar y experimentar en los diferentes motores, es un fragmento de una canción. Publicada originalmente por *Parni Valjak* en 1994
Se seleccionó dicho fragmento debido a que el mismo presenta vocabulario avanzado, uso de preposiciones y declinaciones. Estás ultimas siendo recursos lingüísticos propios del croata y no existen en el inglés. Por esto mismo se espera, que los motores de TA presenten dificultades para alcanzar una traducción de calidad.

Tabla 3

Texto a traducir hr>en

Segmento original	Lengua meta
Ne pitaj me noćas ništa pusti me da šutim Ja noćas trebam mir Stare rane opet peku moje bitke dalje teku, dušo Ti nemaš ništa s tim Sa tvojeg izvora moja se duša napila Žedna tvojih godina I sada mamurna pita gdje je utjeha Gdje je mladost nestala Idu dani ja ih pratim, ponekad do tebe svratim Dušo tražim zaborav Molim sate da se vrate tragovima njenim hodam Tiho kao da je tu Sve još miriše na nju, i dan, i jutro što će doći Nakon ove noći, noći bez sna I dvjesto godina da ih brojim u samoći Otkako je otišla	Don't ask me anything tonight, let me be quiet, I need peace tonight. Old wounds are burning me again, battles inside of me are still ongoing, darling, it has nothing to do with you. From your "well" my soul has gotten drunk, it's been thirsty for your years, and now hangovered it wonders where consolation is, where has the youth disappeared. Days are going by I'm following them, sometimes I drop by to see you, darling, I'm looking for oblivion. I'm praying for those hours to come back, I'm walking her footsteps, silently as if she was here. Everything still smells of her, this day and the morning that will come, after this night, sleepless night, and two hundred years for me to count in loneliness, since she has been gone.

3.3.2 Español > inglés

3.3.2.1 Oraciones, frases y perífrasis

Tabla 4

Listado de oraciones a traducir sp>en

Segmento original	Lengua meta
Vive cerca de la escuela.	He/She lives near the school.
Esto es entre nosotros.	This is between us. / This stays between us.
El domingo no lloverá.	It won't rain on Sunday.
Pasó al lado de su antigua casa	He/She walked by his/her old house. / It happened next to his/her/your/their old house.
El tranvía no funciona por falta de electricidad	The streetcar is not working due to power shortage.
La casa está al lado de la escuela.	The house is by/next to the school.
Los trabajadores recibieron el pago a pesar de la crisis.	The workers got their salaries in spite of the crisis.

Tabla 5

Listado de frases a traducir sp>en

Segmento original	Lengua meta
Yo estoy cantando.	I'm singing.
Sigo leyendo.	I keep reading.
Leo el periódico.	I read the newspaper.
¿Está lloviendo?.	Is it raining?
Ese libro es para mí.	That book is for me.
El pájaro vuela sobre la casa	The bird flies over the house.
Siento que me falta el aire.	I feel like I can't breath. / I feel like I'm short of breath.

Tabla 6

Listado de perífrasis verbales a traducir sp>en

Segmento original	Lengua meta
Ha comenzado a llover.	It's started to rain.
El sol está a punto de salir.	The sun is about to rise/come up.
Estoy leyendo el periódico.	I'm reading the newspaper.
Voy a ir yendo.	I'll get going.
Tienes que comer más.	You have to/need to/must eat more.
Te lo voy a explicar.	I'll explain it to you.

Mañana tengo que ir a París.	I have to go to Paris tomorrow.

Tabla 7

Listado de expresiones idiomáticas a traducir sp>en

Segmento original	Lengua meta
Estar en buenas manos	To be in good hands.
Tener resaca.	To be hangover.
Ser todo oídos.	To be all ears.
¡Vamos!.	Come on! / Let's go!
Meter la pata.	To mess up (big time).
Trabajar de Sol a Sol	To work like a dog/horse.
Estar entre la espada y la pared.	To be between a rock and a hard place.

4 Traducción basada en reglas

4.1.1 Selección del sistema basado en reglas

Para la traducción basada en reglas se decide a utilizar el Software Apertium. Éste forma parte de una plataforma de TA OpenSource, en un principio fue ideada sólo para trabajar con lenguajes que mantengan una misma raíz, pero la misma, gracias a la comunidad, escaló con el tiempo para utilizarse con idiomas no tan cercanos, como por ejemplo el catalán y alemán.

Apertium es una herramienta de código abierto y contiene hoy en día data suficiente para ser usada en 51 pares de lenguas distintas.“La herramienta como primer paso utiliza los

diccionarios cargados para encontrar la palabra equivalente en el idioma objetivo. Luego aplica las reglas de gramática de cada uno de los idiomas para formar la oración coherentemente" (Streiter, Scannell, y Stuflesser, 2006).

Si bien hoy en día la tendencia decrece en la utilización de arquitecturas basadas en reglas, las arquitecturas más primitivas como esta son consideradas para utilizarse en escenarios donde se trabaja con lenguas menos utilizadas o denominadas no centrales Apertium puede utilizarse de manera cloud o local, para este experimento, se instaló Apertium 2.0 y se descargaron los sets de datos disponibles para realizar las pruebas con los idiomas definidos en este trabajo.

4.1.2 Preparación e instalación Apertium

Siguiendo las guías que se encuentran en la Wiki dedicada de Apertium. La herramienta ofrece dos paquetes de instalación para entornos Windows.

- Apertium VirtualBox
- Apertium Simpleton UI

Apertium VirtualBox cuenta con todas las herramientas necesarias para utilizar el traductor y a su vez desarrollar nuevos pares de lenguas, es decir crear los archivos con reglas gramaticales con el fin de desarrollar un nuevo par de lenguas

Apertium Simpleton UI es la herramienta que permite instalar localmente el traductor con los pares de lenguas que se deseen. Sirve para traducir, y no brinda la posibilidad de desarrollar nuevos pares de lenguas.

Para nuestro trabajo de investigación, se opta por Apertium Simpleton UI y se procede a descargar los diccionarios de los pares de lengua a analizar.

La versión Windows se puede descargar de la propia Wiki de la herramienta (http://apertium.projectjj.com/win32/nightly/apertium-simpleton-latest.7z). La primera vez que se ejecuta Apertium Simpleton UI es necesario descargar los pares de lenguajes necesarios para luego poder traducir. Para eso se despliega el instalador de paquetes Apertium.

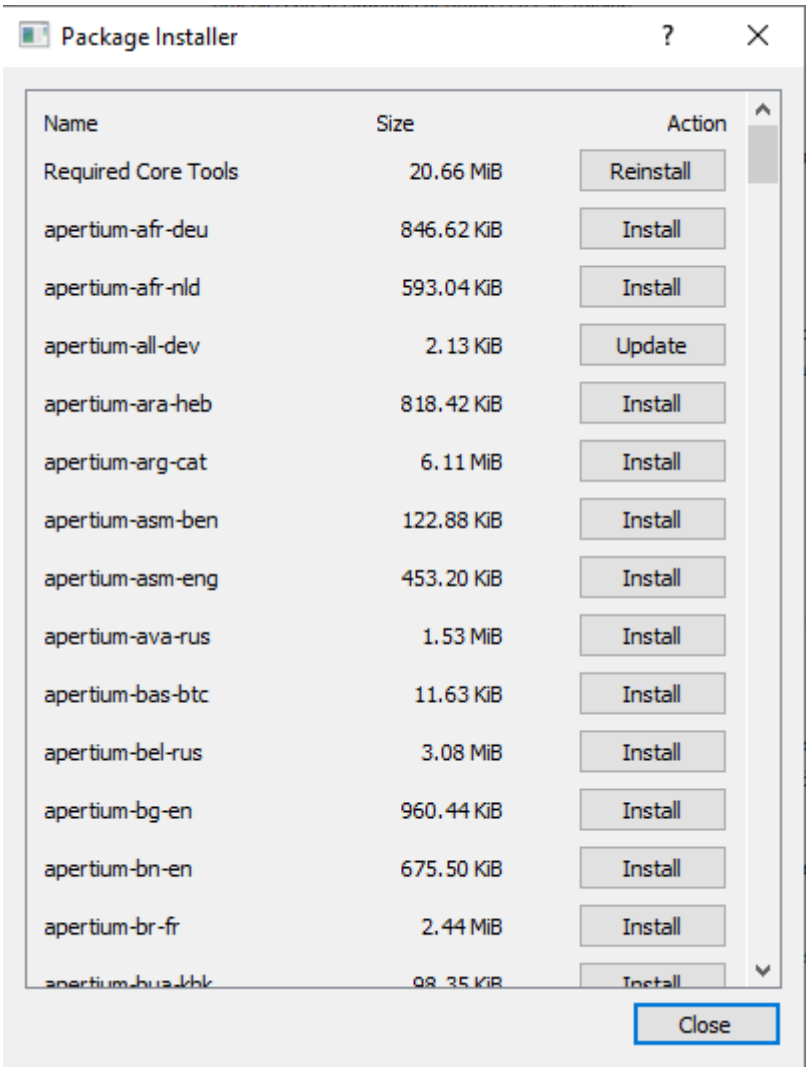

Figura 5 – Paquetes de Apertium

Para el correcto funcionamiento se necesita instalar los siguientes paquetes:

- Required Core Tools
- apertium-hbs-eng (croata > inglés)
- apertium-spa-eng (español > inglés)

De esta manera la herramienta queda lista para su uso y la ejecución del experimento. Apertium Simpleton UI cuenta con dos cuadros de textos y un menú desplegable donde se elige el par de lenguas con el que se quiere trabajar.

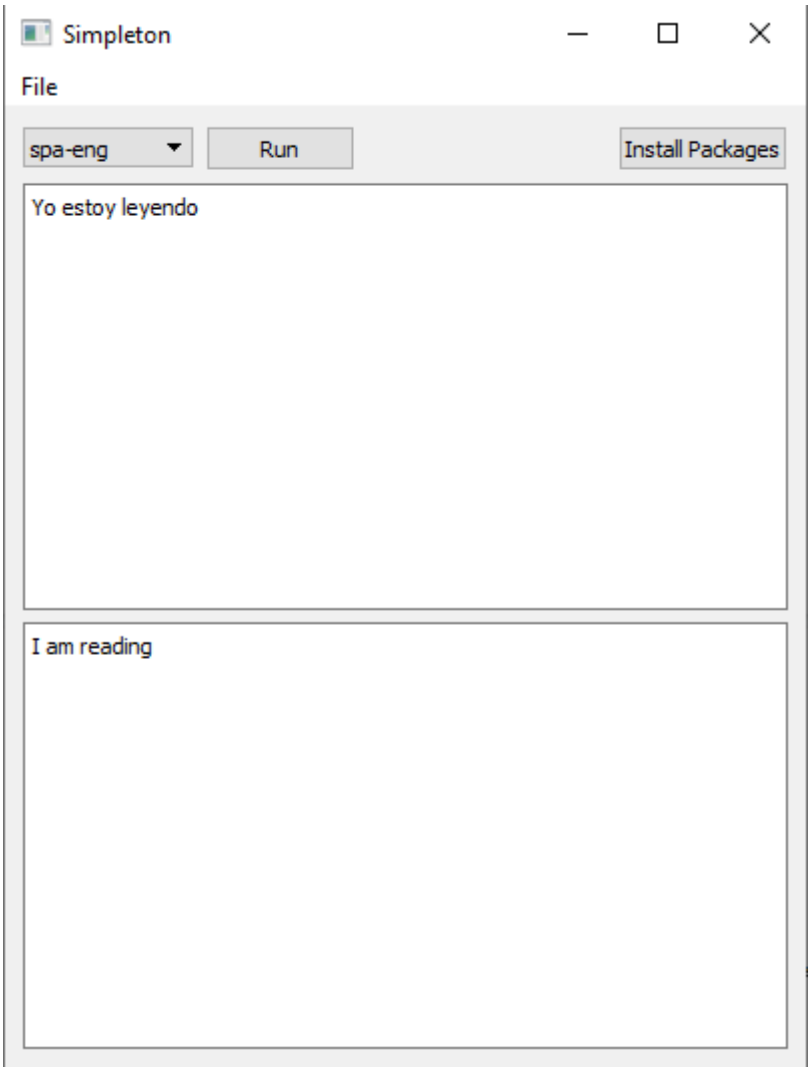

Figura 6 – Interfaz de Usuario Apertium

Una vez seleccionado el par de lenguas deseado a traducir, se ingresa en el recuadro superior el texto en la lengua origen y mediante el botón Run empieza el proceso de traducción mostrando el resultado en el cuadro de texto inferior.

5 Traducción estadística

5.1 Selección del motor estadístico

Como primer paso, se llevó a cabo el análisis de cuatro diferentes herramientas de TAE existentes en el mercado. De las cuatro soluciones analizadas se seleccionan las herramientas que mejor se adecuan al propósito de este trabajo, para llevar a cabo la experimentación Las herramientas analizadas son KantanMT, Moses, Joshua, Bing Translate (Microsoft Translator), Google Colaboratory.

KantanMT es una solución SaaS, con la cual se pueden crear motores de TA y entrenarlos de la manera que el usuario desee. Tiene sus propias herramientas de medición que califica la calidad de las traducciones que se obtienen. Esta herramienta no es gratuita, una vez abonada la licencia correspondiente, permite crear motores y entrenarlos. El usuario no necesita conocimiento técnico para montar un motor y entrenarlo. Al ser una herramienta cloud, tampoco se necesita de grandes recursos computacionales, ya que todas las fases del entrenamiento y ejecución se realizan en la plataforma del proveedor.

Moses es una herramienta OpenSource. Permite que el proceso de punta a punta pueda ser controlado por el usuario. La desventaja que tiene, radica, en que el usuario debe contar con conocimiento técnico para utilizarla. No es una herramienta cloud, por ende, se necesita la capacidad computacional necesaria para alojar y entrenar el motor.

Microsoft Translator, ofrece un motor estadístico gratuito, si se utiliza a través del navegador Bing, la herramienta es cloud y a septiembre 2020 ofrece la posibilidad de traducir entre 73 pares de lenguajes. También ofrece un servicio empresarial que se consume a través de su API privada. Al ser una herramienta gratuita y SaaS, no es posible participar de las fases de entrenamiento, ni creación del motor, sino que solo es posible utilizar el servicio de traducción.

Google Colaboratory es una solución SaaS, de Google, que permite escribir y ejecutar código Python. La misma cuenta con muchas facilidades, especialmente para tareas de aprendizaje automático y análisis de datos. Si bien en este trabajo no es seleccionada debido a limitaciones técnicas en su licencia gratuita. Se la menciona en las futuras líneas de investigación.

Joshua es un traductor automático y estadístico desarrollado en Java. Se distribuye bajo licencia OpenSource, esto quiere decir que es de código abierto. Cuenta con sets de

lenguas pre-construidos para libre utilización. Estos sets de lenguas son modelos previamente entrenados, que se pueden utilizar bajo la licencia de distribución que se ofrece.
Por otro lado, si el usuario desea re-entrenar dicho modelo o crear uno completamente nuevo, también lo puede hacer. Esta herramienta al no ser cloud, también necesita de los recursos computacionales necesarios para realizar el entrenamiento del modelo correspondiente.

La solución seleccionada para llevar a cabo el experimento, es Joshua, debido a que entre las tres, es la única que ofrece un modelo pre-entrenado y tiene una ventaja de performance por sobre Moses. KantanMT no fue elegido, debido a que no cuenta con una licencia académica que permita su utilización.

El servicio de Microsoft Translator también se utiliza para realizar las traducciones correspondientes, pero debido a que no es posible entrenar ni modificar ninguno de los parámetros de su motor, se utilizan sus traducciones como material complementario de investigación

5.2 Recolección de corpus

Con el fin de entrenar el motor estadístico, es necesario primero, recolectar el corpus bilingüe que se utiliza para realizar dicho entrenamiento. Este corpus está conformado por el texto en la lengua origen y la lengua meta. Lo primordial es que el corpus sea lo más extenso y rico posible. También se debe tener en cuenta que cuanto más extenso el corpus es, también más procesamiento computacional se necesita al momento de ejecutar el entrenamiento.

Para recolectar los corpus en los pares de lenguas a trabajar se recurrió a OPUS. OPUS es una colección colaborativa, de textos traducidos y alineados, que tienen el fin de proveer diferentes sets de corpus listos para ser utilizados en el entrenamiento de motores de TA. La particularidad de esta colección es que todo el pre procesamiento de los sets de datos que están disponibles, se hacen de manera automática y por ende no tiene ningún tipo de corrección manual.

Al ser una plataforma colaborativa, la colección crece día a día, por este motivo, es posible encontrar set de datos actualizados y con riqueza de recursos. Para los pares de lenguas a utilizar en este trabajo se decide descargar el corpus *OpenSubtitles v2018*.

Tabla 8

Corpus croata > inglés

Palabras
12655395

Nota: Cantidad de palabras que contiene el corpus hr>en

Tabla 9

Corpus español > inglés

Palabras
11210375

Nota: Cantidad de palabras que contiene el corpus sp>en

5.3 Pre-Procesamiento

Una vez recolectado el corpus que se va a utilizar, se deben realizar una serie de procesos para que dicho corpus pueda ser utilizado en el entrenamiento. Los pasos realizados son:

1. Tokenización.
2. Descarte de oraciones largas.
3. Transformación de minúsculas.

Antes de detallar en qué consisten cada uno de estos pasos, es necesario aclarar que no existe un estándar en la ejecución de dichos procesos. Sino que se realizan en base a la necesidad y al par de lenguas a trabajar. En este escenario se siguen los pasos aconsejados en el mismo sitio de Apache Joshua.

En el paso 1 de Tokenización se busca agregar, entre palabras y los signos de puntuación, espacios, también, si el par de lenguas lo requiere, se dividen las contracciones, por ejemplo, en el caso del inglés la palabra “don´ t” se divide en “don” “t”.

Los métodos de Tokenización varían dependiendo del lenguaje con el que se esté trabajando. En el par de lenguas alcanzadas en este experimento, solo se dividieron contracciones para el idioma inglés.

En el paso 2, se eliminan las oraciones que están conformadas por más de noventa palabras, esto es debido a que para el motor es muy costoso alinearlas al momento del entrenamiento.

En el paso 3 se procesa el texto para eliminar las mayúsculas y transformar todas las letras a minúsculas, esto se hace para reducir el vocabulario. Cabe aclarar que este proceso no se puede realizar en cualquier con cualquier lengua. Ya que, por ejemplo, en el idioma alemán una palabra puede variar su significado de acuerdo a si su primera letra es mayúscula o minúscula.

5.4 Instalación de Joshua

Joshua permite instalar dos versiones de su solución.

- Joshua Language Pack
- Apache Joshua

Joshua Language Pack ofrece modelos pre entrenados listos para su utilización, en este caso lo que se hace es descargar el modelo correspondiente para cada par de lenguas y así es posible realizar la traducción del texto definido. Esta modalidad permite utilizar el modelo estadístico, en un concepto de caja negra, es decir se provee un texto en la lengua origen y se obtiene la traducción, sin necesidad y posibilidad de analizar cómo funciona el motor. La única dependencia que tiene es Java. Cabe aclarar que Joshua se ejecuta bajo el entorno Linux.

Apache Joshua permite tener control de todo el proceso de traducción e incluso crear los propios modelos de lenguas, con esta opción es posible entrenar los modelos ya existentes y de esta manera crear un modelo propio mucho más avanzado.

En este capítulo se describe como instalar y utilizar las dos versiones que ofrece Joshua.

Con Apache Joshua se crea un modelo nuevo a partir del Corpus recolectado, con este modelo creado se genera el Language Pack correspondiente para que cualquier usuario lo pueda utilizar sin necesidad de conocimientos técnicos. Tanto el modelo construido y entrenado en este capítulo, como cualquier modelo pre entrenado que ya ofrece Joshua se pueden utilizar mediante Joshua Language Pack

5.4.1 Modelo Pre-entrenado

5.4.1.1 Joshua Language Pack

Desde la propia Wiki de la herramienta, alojada en Confluence es posible descargar el paquete de lenguas deseado con el modelo pre-entrenado. https://cwiki.apache.org/confluence/display/JOSHUA/Language+Packs. Por ejemplo, en este trabajo, se descargan los siguientes archivos:

- sp-en (español > inglés): apache-joshua-es-en-2016-11-18.tgz
- hr-en (croata > inglés): apache-joshua-hr-en-2016-11-18.tgz

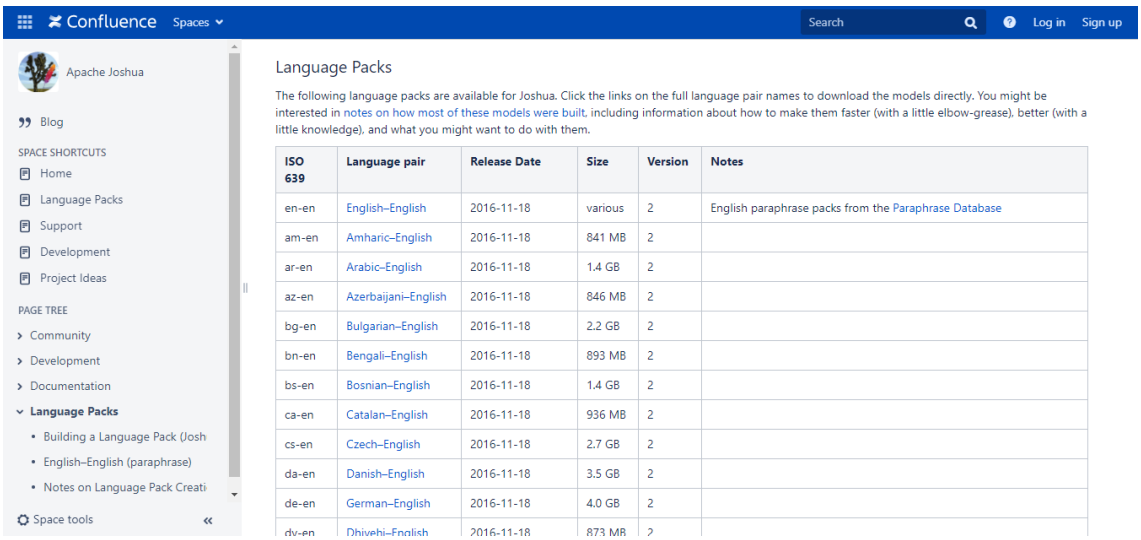

Figura 7 – Confluence Apache Joshua

Una vez descargados los archivos correspondientes al par de lenguas con los que se va a trabajar, se deben ejecutar los siguientes comandos.

Para el par de lenguas español > inglés

```
tar xzf apache-joshua-es-en-2016-11-18.tgz
cd apache-joshua-es-en-2016-11-18
cat exp-es-en.SRC | ./prepare.sh | ./joshua
```

Figura 8 – Comando par de lenguas sp > en

Para el par de lenguas croata > inglés

```
tar xzf apache-joshua-hr-en-2016-11-18.tgz
cd apache-joshua-hr-en-2016-11-18
cat exp-hr-en.src | ./prepare.sh | ./joshua
```

Figura 9 – Comando par de lenguas hr > en

Los archivos *"exp-hr-en.src"* y *"exp-es-en.SRC"* que se utilizan como input, se encuentran conformados por una frase por línea de la lengua origen. De esta manera Joshua toma estos archivos como fuente y realiza la traducción correspondiente.

Se crearon dos archivos para cada par de lenguas. El primero contiene todas las frases a traducir y el segundo contiene el texto corto detallado en el capítulo 3.

También es posible ejecutar la plataforma en modo server, con el siguiente comando:

```
/joshua -server-port 5674 -server-type http
```

Figura 10 – Comando ejecución modo server

De esta manera, abriendo en un navegador web el archivo *"web/index.html?port=5674"* (que provee Joshua al descargarlo) se pueden traducir segmentos de texto a partir de una interfaz más agradable para el usuario. Donde se ingresa el texto origen y se solicita la traducción mediante un botón. De esta manera, cuando la herramienta está implementada y ejecutándose en modo server, la puede utilizar cualquier usuario sin necesidad de tener conocimientos de Linux.

5.4.2 Creación de un modelo nuevo

5.4.2.1 Apache Joshua

En el punto anterior, se describe como ejecutar un modelo previamente entrenado. Aquí, se detalla cómo crear un modelo desde cero y como paquetizarlo para que pueda ser ejecutado por terceros mediante Joshua Language Pack.

Cuando el pre-procesamiento finaliza, empieza la etapa de entrenamiento del modelo, en esta etapa se parametriza de qué manera se quiere entrenar al modelo. A continuación, se explica los valores y parámetros definidos.

5.4.2.2 Alineación

En una de las fases del entrenamiento, Joshua realiza la alineación, donde se mapean las oraciones de un idioma a otro, para realizar dicho proceso, se usa la herramienta GIZA++. Esta herramienta funciona de manera nativa y se incluye en el paquete instalador Apache Joshua.

El proceso de alineación es uno de los más costosos, debido, a que el entrenamiento consta de varias iteraciones en las que, en cada una de ellas se recalcula la mejor posibilidad de alineación para las palabras. El tiempo que tarda es directamente proporcional a la extensión del corpus.

5.4.2.3 Modelo de n-gramas

Para entrenar el modelo de lenguaje de n-gramas se utiliza la herramienta KenLM. Ésta brinda la posibilidad de elegir el valor de n. De manera default está parametrizado en 5. Cuanto mayor sea el valor de n se obtendrá a final de cuentas una mejor traducción, sin embargo, también significa que se consumen más recursos haciendo el proceso mucho más lento.

Para el caso de estudio presentado aquí se parametriza en 4 gramas. También por otro lado para entrenar un modelo de lenguaje 5 gramas se necesita un corpus o memoria de traducción mucho mayor al disponible.

5.4.2.4 Instalación y entrenamiento

Como primer paso es necesario instalar Apache Joshua clonándolo desde el repositorio de GitHub

```
1  git clone https://github.com/apache/incubator-joshua joshua
2
3
4
```

Figura 11 – Comando clonado Apache Joshua

Configurar el directorio raíz instalar Joshua y sus dependencias (Hadoop y KenLM)

```
1  cd joshua
2  export JOSHUA=$(pwd)
3  # add this to your ~/.bashrc, too
4  echo "export JOSHUA=$JOSHUA" >> ~/.bashrc
5
6  # compile Joshua, run tests, and build the jar file
7  mvn package
8
9  # Download dependencies used for building and running models
10 bash download-deps.sh
11
```

Figura 12 – Comando instalación Apache Joshua

En este punto se crea un directorio donde dejaremos los archivos que corresponden al corpus, el set de datos usados para la etapa de validación y tunning. En este caso vamos a limitar parametrizando que Joshua tenga en cuenta oraciones de hasta 90 palabras en ambas lenguas, luego configuraremos para que se ejecuten 3 iteraciones del paso de tunning.

De esta manera se ejecuta el pipeline para construir el motor y entrenarlo con la data que le proveemos de nuestro corpus. utiliza los valores default tanto para el alineador GIZA++ como para crear el modelo de lenguaje con LM, en este último hemos configurado para que se trabaje en 4 gramas. Con el siguiente comando, vamos a crear un nuevo motor basado en frases

```
$JOSHUA/bin/pipeline.pl \
  --rundir 2 \
  --readme "Baseline phrase run" \
  --source hr \
  --target en \
  --type phrase \
  --corpus $MILICH/corpus/ldc/hr_en_train \
  --tune $MILICH/corpus/ldc/hr_en_dev \
  --test $MILICH/corpus/ldc/hr_en_dev2 \
  --maxlen 9 \
  --maxlen-tune 90 \
  --maxlen-test 90 \
  --tuner-iterations 3 \
  --lm-order 4

```

Figura 13 – Comando creación motor basado en frases

Una vez que se ejecutan los comandos, se crea el modelo y se entrena el mismo. Solo resta crear el Language Pack para que cualquier usuario pueda utilizar y ejecutar el motor de la misma manera que se detalla en la sección Joshua Language Pack. Para crear el Language Pack se ejecuta el script que se detalla:

```
$JOSHUA/scripts/language-pack/build_lp.sh.


```

Figura 14 – Comando creación Language Pack

Este script se encarga de juntar los archivos necesarios para crear el paquete ejecutable a partir del modelo recientemente creado. Si se desea cambiar el nombre del paquete a generar es necesario detallarlo dentro del script. En este caso se siguió la configuración default que provee Joshua y se generó el paquete con el siguiente nombre `apache-joshua-<LANGPAIR>-YYYY-MM-DD/.`

Dicho archivo está listo para ejecutarse de la misma manera que cualquier Language Pack de Joshua. El proceso completo de entrenamiento y traducción para ambos pares de idiomas se llevó a cabo en 127 horas.

6 Traducción Neuronal

6.1 Selección del motor neuronal

Para el desarrollo del experimento utilizando motores de TAN, se eligieron dos herramientas para trabajar

- OpenNMT
- DeepL

OpenNMT: Es un motor de traducción neuronal con la característica de ser OpenSource y trabaja con diferentes implementaciones de arquitecturas neuronales. La plataforma ofrece la posibilidad de trabajar con dos frameworks de Deep Learning diferentes, PyTorch y TensorFlow. Para el caso de esta investigación se decide utilizar PyTorch. La particularidad de esta implementación es que maneja una interfaz de usuario amigable y no necesita de software de terceros para poder generar un motor de TA, también permite aprovechar la capacidad de procesamiento del GPU para realizar el entrenamiento y posterior traducción. En la TAN es necesario entrenar al motor con un CORPUS, al igual que con la TAE. Debido a que el entrenamiento de un motor Neuronal requiere de mucha capacidad computacional, se va a complementar el experimento de investigación utilizando el servicio de Google Translate.

DeepL es un motor de traducción que desde el año 2016 implementó TAN, actualmente puede trabajar con 109 pares de lenguas, de los cuales 27 utilizan TAN. Dentro de los 27 pares de lenguas, se encuentran las alcanzadas por este trabajo de investigación.

Para la metodología de este experimento se realizan las traducciones con DeepL y se analizan los resultados de igual forma que con OpenNMT, de esta manera se logra complementar la investigación de igual forma que se realizó con Microsoft Translator en el capítulo de traducción estadística. La versión gratuita de este motor permite traducir hasta 5000 caracteres por ejecución. No es posible entrenar este motor, ya que solo se tiene acceso al mismo como solución SaaS.

6.2 Corpus

OpenNMT cuenta con modelos pre-entrenados, para el par de lenguas español – inglés. Sin embargo, se usan nuevamente los corpus utilizados con el motor TAE. De esta manera al obtener los resultados podremos analizar las traducciones de diferentes motores, entrenados con los mismos corpus. De la misma manera que se realizó con el motor TAE, los corpus deben ser tokenizados. Para definir la arquitectura e hiperparámetros se siguen las configuraciones definidas por defecto brindadas por OpenNMT

El motor de TAN necesita de corpus de validación, estos son usados para evaluar la convergencia del entrenamiento, lo normal es que no contengan más de 5000 líneas. De esta manera se extrae un fragmento del corpus paralelo original para generar los corpus de validación.

6.3 Sesgo y Sobreajuste

Las dos técnicas utilizadas para mitigar la recurrencia de sesgo y sobreajuste son:

- Early Stopping
- DropOut

Para aplicar Early Stopping se elige al azar un 10% de frases del corpus paralelo para validación. Cuando se detecta que la métrica Bleu desciende y la función de error de validación cruzada asciende para el corpus de validación, se detiene el ajuste. Se utiliza la técnica DropOut usando la configuración sugerida por OpenNMT siendo de 0.1 respectivamente.

6.4 Instalación y ejecución de OpenNMT

Para instalar, crear y entrenar el motor neuronal se utilizó una placa de procesamiento gráfico (GPU), ya que, sin ella, la fase de entrenamiento duraría semanas. El motor se ejecuta en ambiente Linux, y para ello se creó una máquina virtual en Windows 10 utilizando Ubuntu. Para instalar OpenNMT se ejecuta el siguiente comando

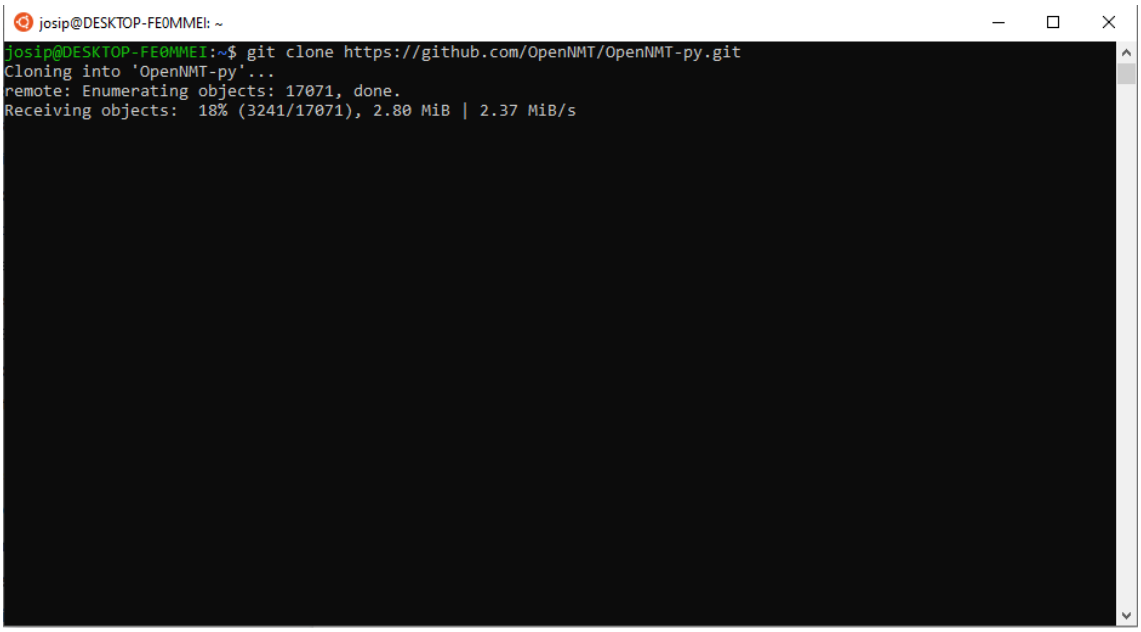

Figura 15 – Comando Instalación OpenNMT

Una vez instalado OpenNMT, es necesario generar un archivo YAML de configuración, donde se parametriza y define la data que se utilizará y los paths para las memorias de traducción basándose en las configuraciones sugeridas por OpenNMT

```yaml
# motor_hr_en.yaml

## Acá se define donde serán guardadas las muestras
save_data: motor-hren/run/example
## Path donde se crean las memorias de traducción
src_vocab: motor-hren/run/example.vocab.src
tgt_vocab: motor-hren/run/example.vocab.tgt
# Comando para no permitir sobrescribir archivos
overwrite: False

# Path donde se encuentran el corpus original y el de validación
# Corpus opts:
data:
    corpus_1:
        path_src: motor-hren/src-train.txt
        path_tgt: motor-hren/tgt-train.txt
    valid:
        path_src: motor-ende/src-val.txt
        path_tgt: motor-ende/tgt-val.txt
...
```

Figura 16 – Configuración archivo YAML

Con esta configuración ya es posible construir las memorias de traducción que se usarán para entrenar el modelo. Y esto se realiza mediante el siguiente comando.

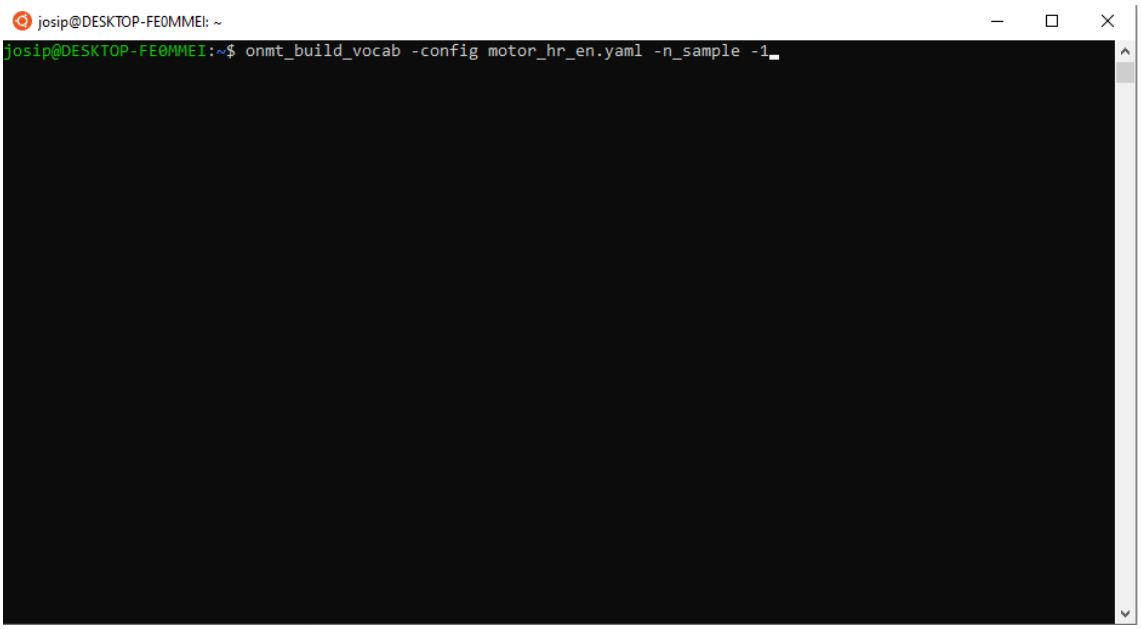

Figura 17 – Comando construcción memorias

El comando -n_sample hace referencia al número de líneas de cada corpus que se utiliza para construir las memorias de traducción. Para este caso utilizamos -1, de esta manera se utiliza el corpus completo El siguiente paso es entrenar el modelo, para eso es necesario agregar la siguiente información al archivo de configuración YAML

- Hiperparametros específicos del entrenamiento
- El path de las memorias que se utiliza (el creado en el paso anterior mediante el comando onmt_build_vocab)

```yaml

# motor_hr_en.yaml

...

# Memorias de traducción que fueron creadas
src_vocab: motor-hren/run/example.vocab.src
tgt_vocab: motor-hren/run/example.vocab.tgt

# Acá se define cuantas GPU se utilizan
world_size: 1
gpu_ranks: [0]

# Donde guardar los puntos de control
save_model: motor-hren/run/model
save_checkpoint_steps: 500
train_steps: 1000
valid_steps: 500

# Model
encoder_type: transformer
decoder_type: transformer
enc_layers: 2
dec_layers: 2
heads: 8
rnn_size: 500
word_vec_size: 500
transformer_ff: 2048
dropout_steps: [0]
dropout: [0.3]
attention_dropout: [0.1]
share_decoder_embeddings: true
share_embeddings: true

```

Figura 18 – Comando hiperparámetros en YAML

Una vez parametrizado el archivo YAML, se ejecuta el siguiente comando para entrenar el modelo.

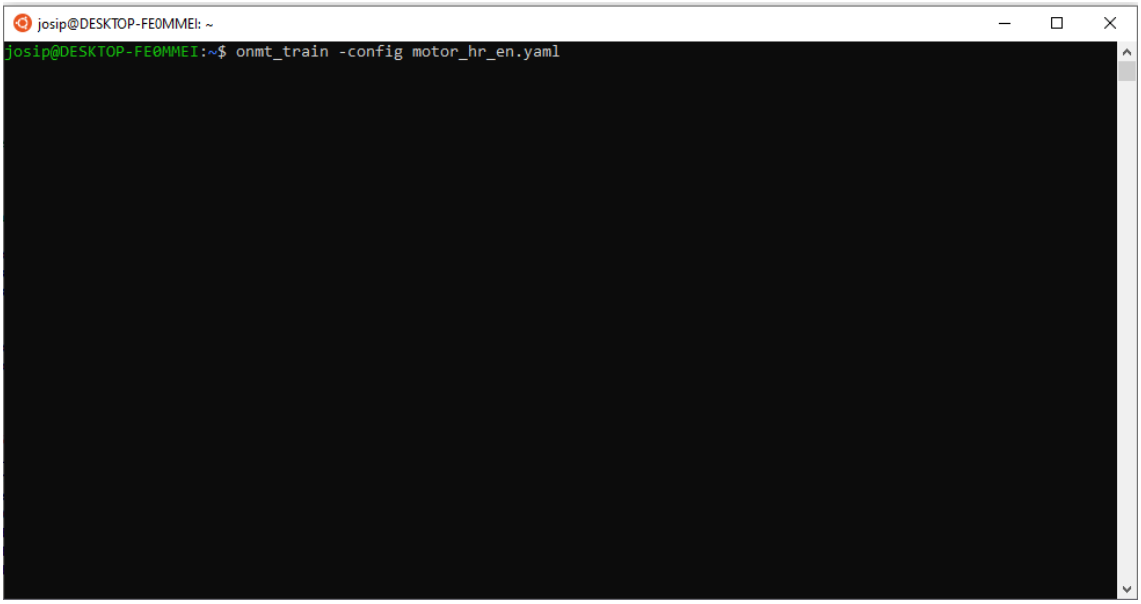

Figura 19 – Comando entrenamiento modelo

Esta configuración ejecuta el modelo por defecto, que consta de 2-capas LSTM con 500 unidades ocultas tanto en el codificador como el decodificador. Y se está ejecutando en una sola GPU. Una vez que termina el entrenamiento, es posible empezar a traducir mediante el siguiente comando.

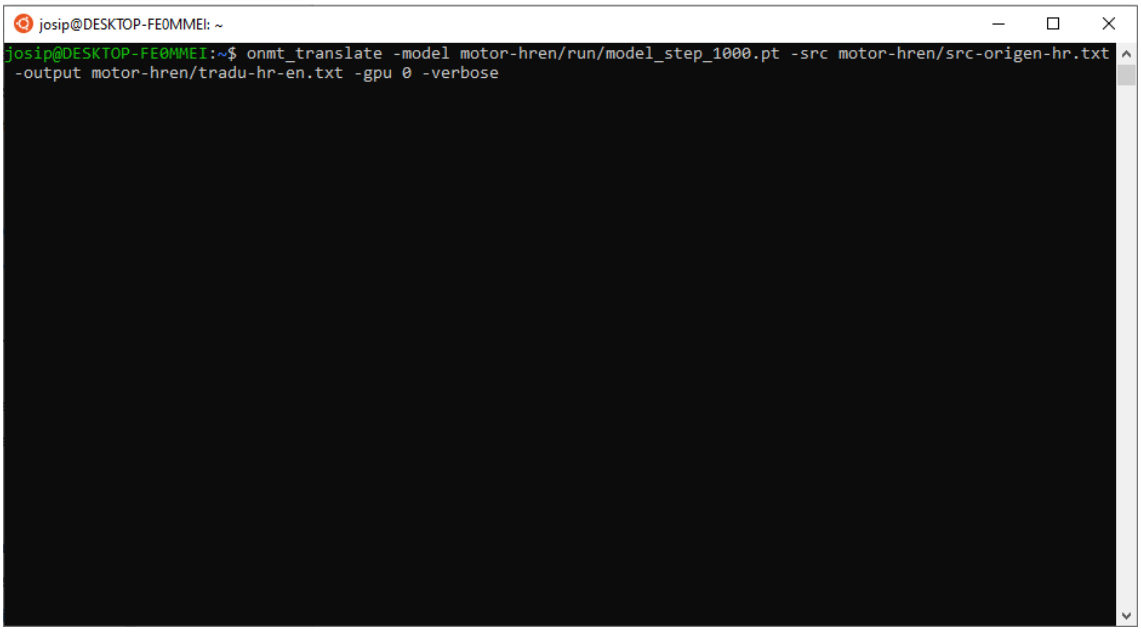

Figura 19 – Comando uso de traductor

En archivo src-origen-hr.txt se encuentra el texto a traducir y las traducciones quedan generadas en el archivo tradu-hr-en.txt. El nombre del modelo tiene el siguiente formato "model_step_1000" donde 1000 es el valor que indica la cantidad de steps definidos en el parámetro train_steps.

Una vez finalizada la creación y entrenamiento del motor croata > inglés, se procede a repetir el procedimiento para crear y entrenar el motor español > inglés

Archivo YAML para motor sp > en

```yaml
# motor_sp_en.yaml

## Acá se define donde serán guardadas las muestras
save_data: motor-spen/run/example
## Path donde se crean las memorias de traducción
src_vocab: motor-spen/run/example.vocab.src
tgt_vocab: motor-spen/run/example.vocab.tgt
# Comando para no permitir sobrescribir archivos
overwrite: False

# Path donde se encuentran el corpus original y el de validación
# Corpus opts:
data:
    corpus_1:
        path_src: motor-spen/src-train.txt
        path_tgt: motor-spen/tgt-train.txt
    valid:
        path_src: motor-spen/src-val.txt
        path_tgt: motor-spen/tgt-val.txt

# Memorias de traducción que fueron creadas
src_vocab: motor-spen/run/example.vocab.src
tgt_vocab: motor-spen/run/example.vocab.tgt

# Acá se define cuantas GPU se utilizan
world_size: 1
gpu_ranks: [0]

# Donde guardar los puntos de control
save_model: motor-spen/run/model
save_checkpoint_steps: 500
train_steps: 1000
valid_steps: 500

# Model
encoder_type: transformer
decoder_type: transformer
enc_layers: 2
dec_layers: 2
heads: 8
rnn_size: 500
word_vec_size: 500
transformer_ff: 2048
dropout_steps: [0]
dropout: [0.3]
attention_dropout: [0.1]
share_decoder_embeddings: true
share_embeddings: true
```

YAML Ain't Markup Language

Figura 20 – Archivo YAML sp > en

Comando para construir memorias de traducción español > inglés

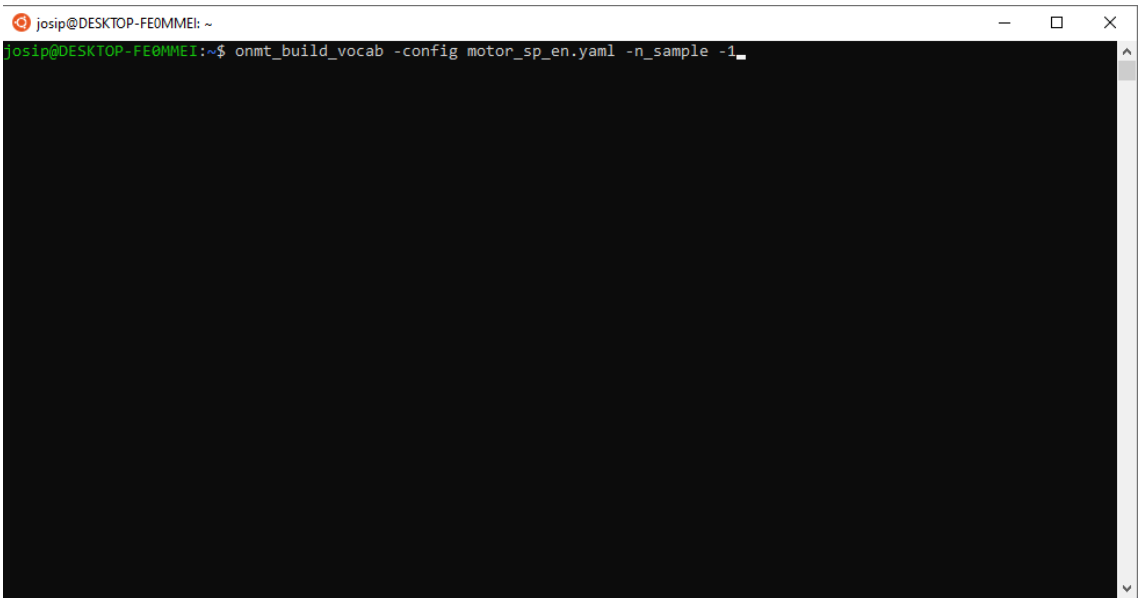

Figura 22 – Comando construcción memorias sp > en

Comando para entrenar el modelo español > inglés

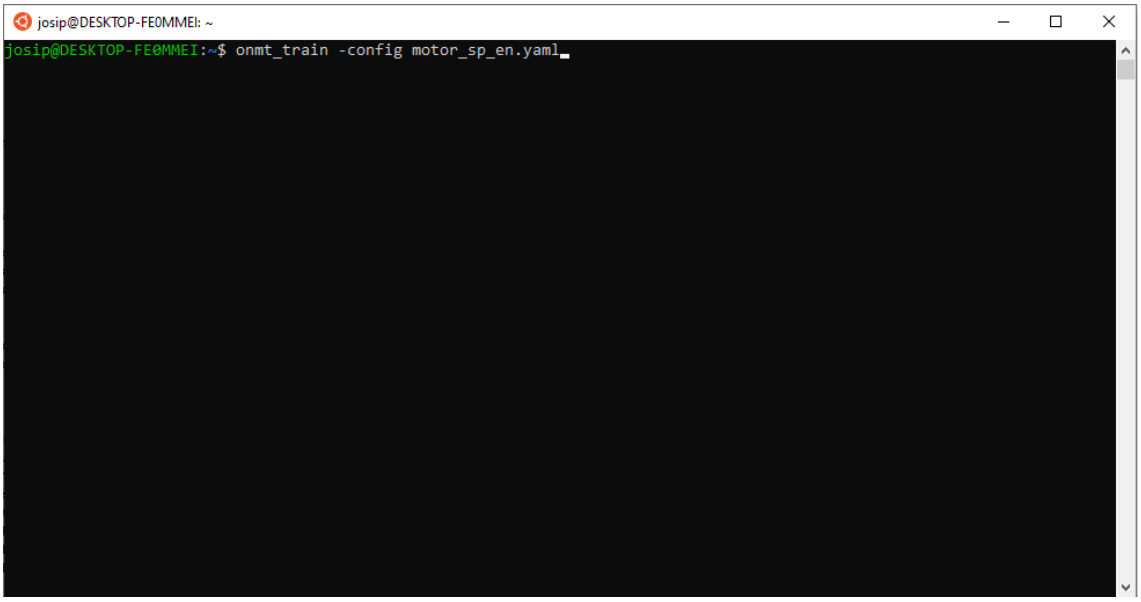

Figura 23 – Comando entrenamiento modelo sp > en

Comando para traducir

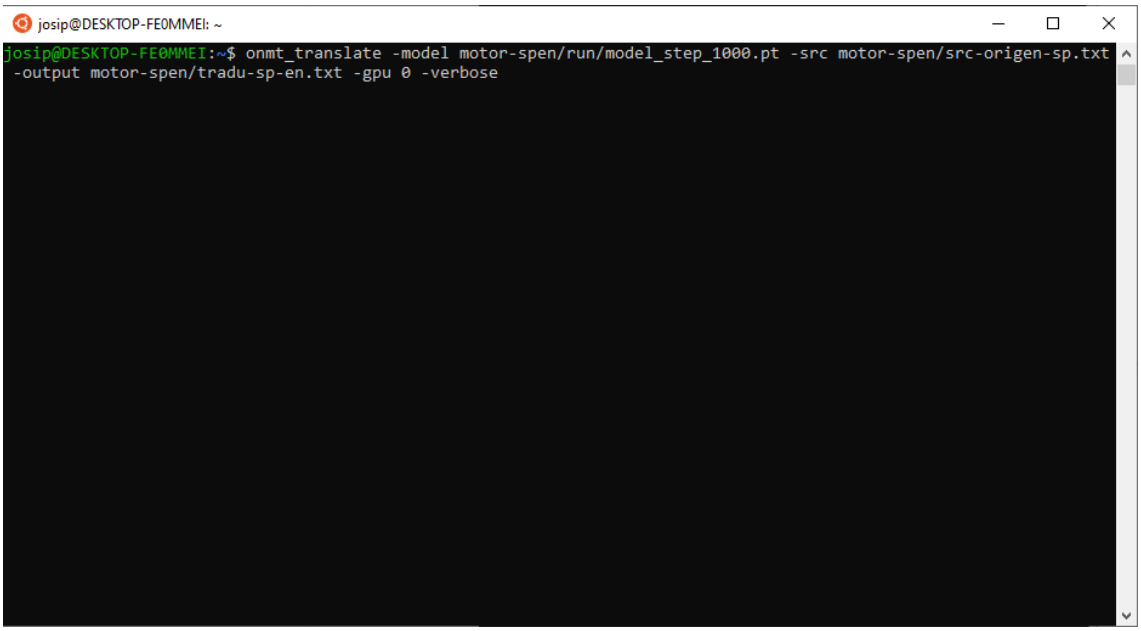

Figura 24 – Comando traducción en modelo sp > en

En el archivo src-origen-sp.txt se encuentra el texto a traducir y las traducciones quedan generadas en el archivo tradu-sp-en.txt El proceso completo de entrenamiento y traducción para ambos pares de idiomas se llevó a cabo en 197 horas.

7 Herramientas

7.1 Introducción

Los avances tecnológicos son un pilar clave en el campo de la traducción profesional. Tal es así que, en la última década se desarrollaron múltiples herramientas para facilitar y ayudar el trabajo del traductor. Como toda tecnología, las diferentes herramientas presentan opiniones encontradas acerca de cómo infieren en el rol específico del profesional. Es por eso que en este capítulo se realiza un sondeo de campo entre profesionales y estudiantes, con el fin de entender cuál es el grado de conocimiento que los traductores tienen por sobre los avances tecnológicos en el rubro, cuáles son las herramientas que se utilizan y cuál es la eficacia de las mismas al usarlas para traducir textos de carácter tanto científico como literario.

7.2 Objetivo

El objetivo de este capítulo es recopilar la opinión de los profesionales y estudiantes de la carrera de traducción, acerca de las herramientas tecnológicas utilizadas y disponibles en el rubro.

Para llevar a cabo este objetivo, se diseñó una encuesta que es utilizada para recolectar datos y realizar un análisis descriptivo de los mismos.

7.3 Encuesta

7.3.1 Fases de la traducción

Si bien las tareas del profesional no se limitan solo al momento de realizar la traducción propiamente dicha, sino que también, existen varios procesos que infieren en el circuito completo de la traducción. Basándonos en Olalla - Soler (2013), el siguiente diagrama de tareas, describe las fases básicas del proceso de traducción humano. Si bien dependiendo de las necesidades, algunas de éstas pueden ser omitidas.

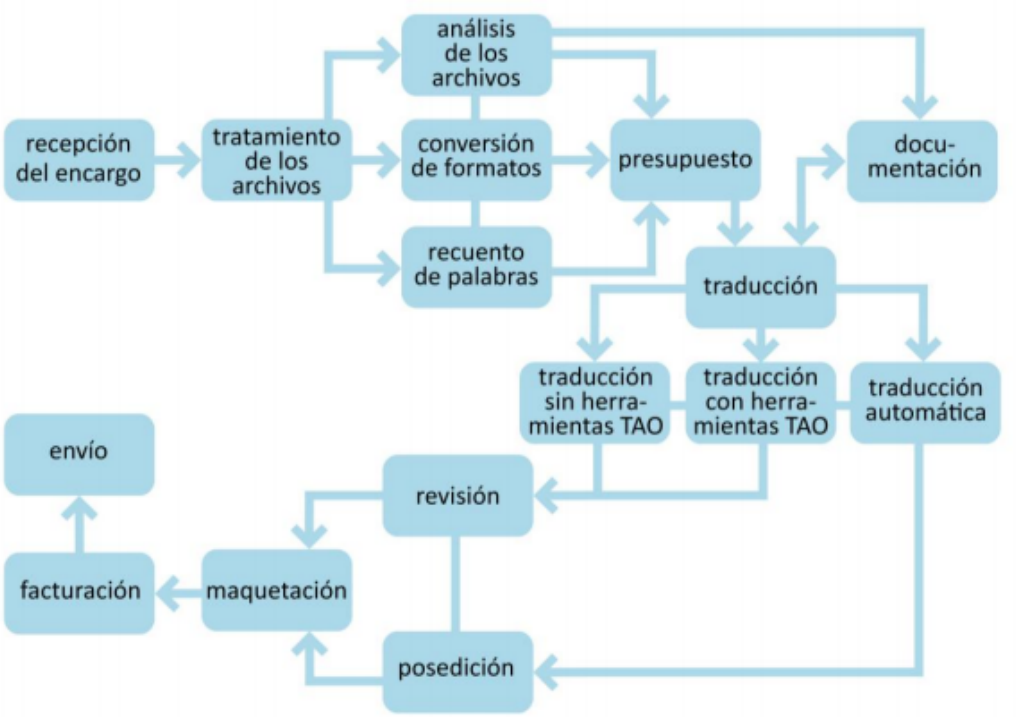

Figura 25 – Fases en el proceso de traducción humana

Como bien se ejemplifica, existen varias fases en cada una de las tareas. Esta encuesta apunta a recolectar datos de las tecnologías conocidas y utilizadas en las fases de, tratamiento de archivos, traducción, revisión y post-edición. En el año 2014 se introdujo un modelo de trabajo actualizado, donde se agrupan las tareas de manera transversal.

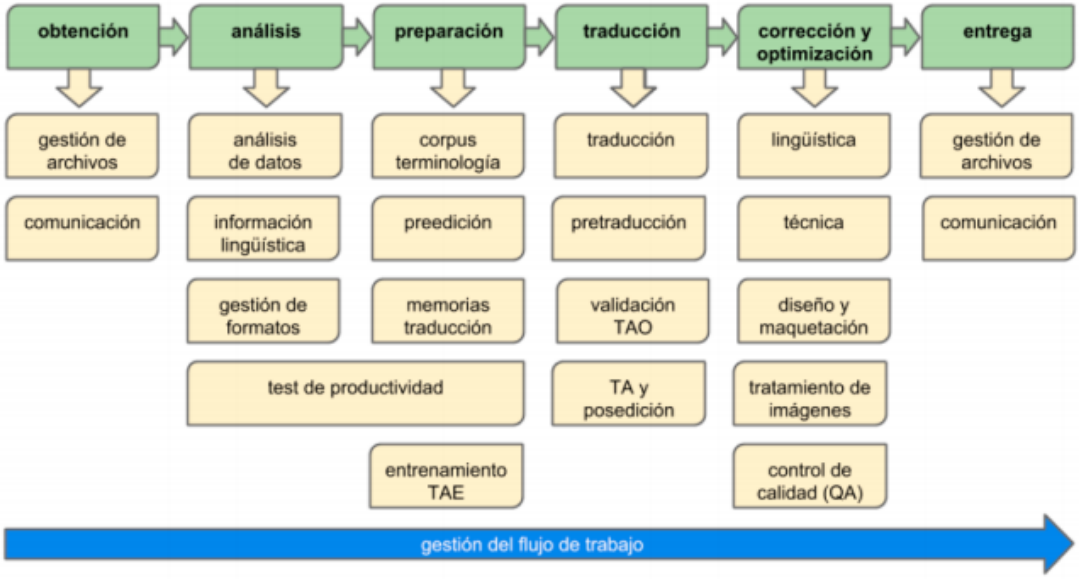

Figura 26 – Fases en el proceso de traducción humana versión II

Este modelo, publicado por (Martín-Mor, Piqué y Sánchez-Gijón, 2014) si bien difiere en su diseño, las tareas que se abordan en esta encuesta se encuentran agrupadas dentro de las fases de preparación y traducción.

7.3.2 Universo

El universo de estudio son los estudiantes y profesionales de las carreras de traducción. La muestra se conforma por un total de 154 personas de las cuales 105 son egresados y 49 estudiantes. La institución donde se llevó a cabo la encuesta es la Escuela Normal Superior en Lenguas Vivas "Sofía E. Broquen de Spangenberg".

7.3.3 Variables

La variable independiente establecida, es el ciclo en el que se encuentra el encuestado, se dividen entre estudiantes (grado y pre grado) y egresados. Dentro de las variables dependientes se encuentran

- La valoración de los conocimientos sobre TA.
- La ponderación de programas y planes de estudios que contengan contenidos referidos a la traducción automática.
- Importancia y uso de TA.
- Herramientas aplicadas en las fases de TA.
- Análisis de eficiencia de cada una de las herramientas de TA expuestas.

7.3.4 Diseño de la encuesta

La encuesta se dividió en las siguientes secciones

1. Traducción Automática en el contexto académico.
2. Eficacia de las herramientas informáticas de traducción desde la perspectiva del traductor científico - literario.
3. Uso y conocimientos de herramientas y recursos disponibles.

En la sección 1 los sujetos deben detallar, en las etapas formativas de la carrera, si tuvieron acceso, a través de su institución, a herramientas de TA. En la sección 2 se busca entender cuál es la opinión sobre los motores de traducción automática que existen en la actualidad desde la perspectiva del profesional traductor. En la sección 3 se busca listar cuales son los recursos que se utilizan, cuáles se conocen y cuáles se desconocen por completo, para ello se brindan nombres de diversas herramientas, basándose en su nombre comercial, se listan las principales herramientas de acceso gratuito y pago.

7.3.5 Índices y Subíndices

Los ítems de la sección 1 y 2 se organizan en índices y subíndices. De esta manera y a través de la escala de Likert se puede obtener una puntuación detallada de todos los elementos que se buscan medir. La puntuación de cada subíndice va de 1 a 7. La suma de todos los subíndices da como resultado el valor obtenido para el índice correspondiente a cada sección.

Composición de índices por sección

Tabla 8

Sección contexto académico

Índice Sección 1	Puntuación
Traducción Automática en el contexto académico	5 - 35
Subíndice	
1.1 Como estudiante o profesional estoy familiarizado con las herramientas de T.A. (Traducción Estadística, Traducción Neuronal, Traducción basada en Reglas, o cualquier otro tipo de herramienta de TA).	1 - 7
1.2 Como estudiante o profesional, creo que las herramientas de Traducción Automática deben ser un contenido esencial en el plan de estudios.	1 - 7
1.3 Como estudiante, estuve en contacto con materias en las cuales se abordó el uso y funcionamiento de herramientas de TA (cualquier tipo).	1 - 7
1.4 Como estudiante o profesional, creo que las herramientas de TA están sobrevaloradas.	1 - 7
1.5 Como estudiante o profesional, creo que es importante dominar las herramientas de TA.	1 - 7

Tabla 9

Sección eficacia de herramientas

Índice Sección 2	Puntuación
Eficacia de herramientas informáticas	7 - 49
Subíndice	
2.1 Cuando traduzco, utilizo herramientas de TA (Basada en reglas, estadística, neuronales o cualquier tipo).	1 - 7
2.2 Como estudiante o profesional, creé o entrené motores de traducción automática.	1 - 7
2.3 Como estudiante o profesional participe del diseño, parametrización o realice la alineación de memorias de traducción utilizadas como corpus para entrenar motores de TA.	1 - 7
2.4 Como estudiante o profesional, obtuve resultados positivos al implementar el uso de motores de TA.	1 - 7
2.5 Como estudiante o profesional, considero que la creación de motores de TA consume una cantidad de tiempo demasiada alta para los resultados que se obtienen.	1 - 7
2.6 Como estudiante o profesional, considero que las herramientas de TA están facilitando las tareas del traductor.	1 - 7

2.7 Como estudiante o profesional considero que la inteligencia artificial puede llegar a tener en cuenta variables como el contexto o la cultura para realizar una traducción.	1 - 7

En la sección 3 los encuestados, deben mencionar cuales son los recursos y herramientas que conocen o utilizaron al menos una vez dentro del ámbito profesional o académico. Se ofrece un listado con las principales herramientas del mercado, más la posibilidad de que el encuestado pueda detallar herramientas no listadas

Los recursos definidos son:

Motores de Traducción Automática

OpenNMT, KantanMT, Moses, Joshua, Microsoft Translator, Google Translate, Moses, SDL Trados Studio, MemoQ, Wordfast, OmegaT, Virtaal, Across, Dejavu, Google Translate, Pairapharase, DeepL

Herramientas de Corrección

TQAuditor, ChangeTracker, ApSicComparator, ApSicXBench

Herramientas de alineación

Abby Aligner, WinAlign, GIZZA, ParaConc, Bitext2mx, MultiTrans, SDLX

7.3.6 Escalas

Para los ítems donde se usa la escala de Likert, se utiliza una escala de cinco niveles para analizar a interpretar los resultados de cada sección. En los cinco niveles se agrupan los datos de la siguiente manera:

- Valoración muy baja
- Valoración baja
- Valoración media
- Valoración alta

- Valoración muy alta

En la siguiente tabla se detallan las escalas de valoración para Índices y Subíndices

Tabla 10

Valoraciones de sección 1

Sección 1	Índice	Subíndice
Valoración muy baja	5 - 11	1 - 2,19
Valoración baja	11,01 - 17	2,20 - 3,39
Valoración media	17,01 - 23	3,40 - 4,59
Valoración alta	23,01 - 29	4,60 - 6,79
Valoración muy alta	29,01 - 35	6,80 – 7

Tabla 11

Valoraciones de sección 2

Sección 2	Índice	Subíndice
Valoración muy baja	7 - 15.49	1 - 2,19
Valoración baja	15,50 - 23,94	2,20 - 3,39
Valoración media	23,95 - 32,40	3,40 - 4,59
Valoración alta	32,41 - 40,82	4,60 - 6,79
Valoración muy alta	40,83 - 49	6,80 - 7

En la sección 3 no se utiliza ninguna escala específica, sino que se detallan los resultados agrupados por tipo de herramienta con los gráficos de los resultados obtenidos.

7.3.7 Resultados

Tabla 12

Resultados sección 1 Total

Índice Sección 1	Puntuación
Total	24.71 - Alta
Subíndice	
1.2	5.21 - Alta
1.4	5.97 - Alta
1.5	3.41 - Media
1.6	3.84 - Media
1.7	6.28 - Alta

Tabla 13

Resultados sección 1 Egresados

Índice Sección 1	Puntuación
Total	24.43 - Alta
Subíndice	
1.2	5.37 - Alta
1.3	5.83 - Alta
1.4	2.94 - Baja
1.5	4.10 - Media
1.6	6.19 - Alta

Tabla 14

Resultados sección 1 Estudiantes

Índice Sección 1	Puntuación
Total	25.33 - Alta
Subíndice	
1.2	4.85 - Alta
1.3	6.27 - Alta
1.4	4.44 - Media
1.5	3.29 - Baja

1.6	6.48 - Alta

Tabla 15

Resultados Sección 2 Total

Índice Sección 2	Puntuación
Total	25.69 - Media
Subíndice	
2.1	4.65 - Alta
2.2	2.28 - Baja
2.3	2.27 - Baja
2.4	4.29 - Media
2.5	3.77 - Media
2.6	5.29 - Alta
2.7	3.14 - Baja

Tabla 16

Resultados Sección 2 Egresados

Índice Sección 2	Puntuación
Total	26.02 - Media
Subíndice	
2.1	4.64 - Alta
2.2	2.40 - Baja
2.3	2.40 - Baja
2.4	4.30 - Media
2.5	3.93 - Media
2.6	5.21 - Alta
2.7	3.14 - Baja

Tabla 17

Resultados Sección 2 Estudiantes

Índice Sección 2	Puntuación
Total	24.96 - Media
Subíndice	
2.1	4.69 - Alta
2.2	2.02 – Muy Baja
2.3	1.98 – Muy Baja
2.4	4.25 - Media
2.5	3.42 - Media
2.6	5.48 - Alta
2.7	3.12 - Baja

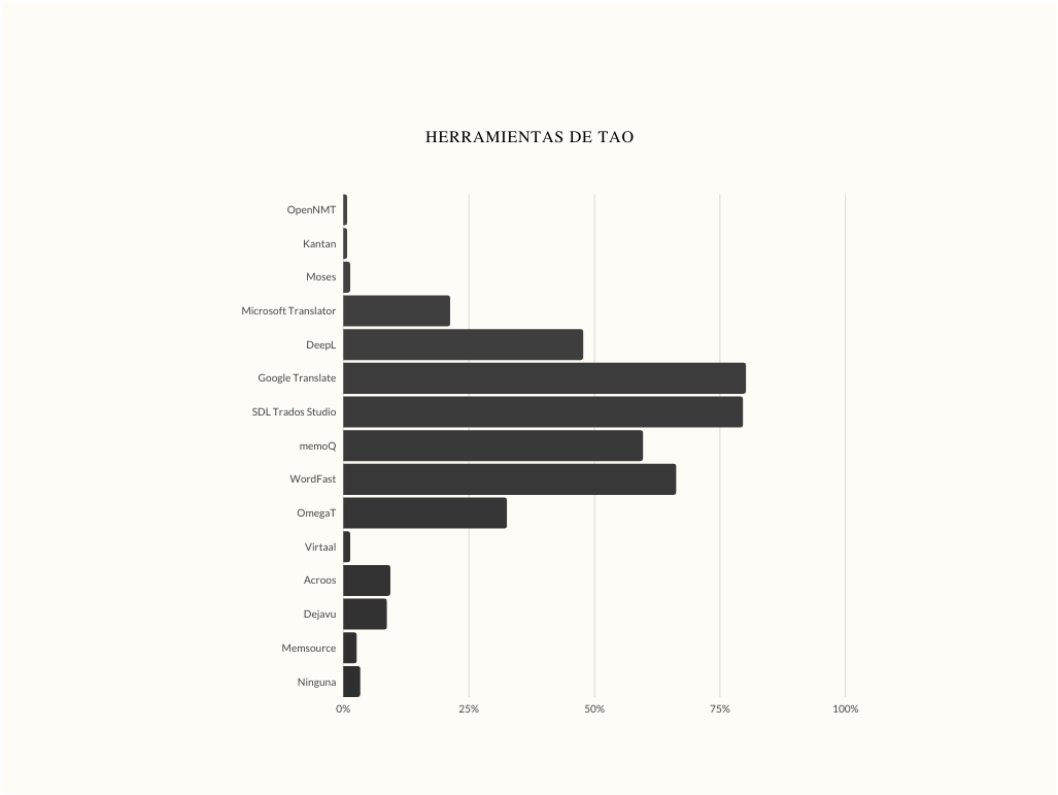

Figura 27 – Gráficos herramientas TAO

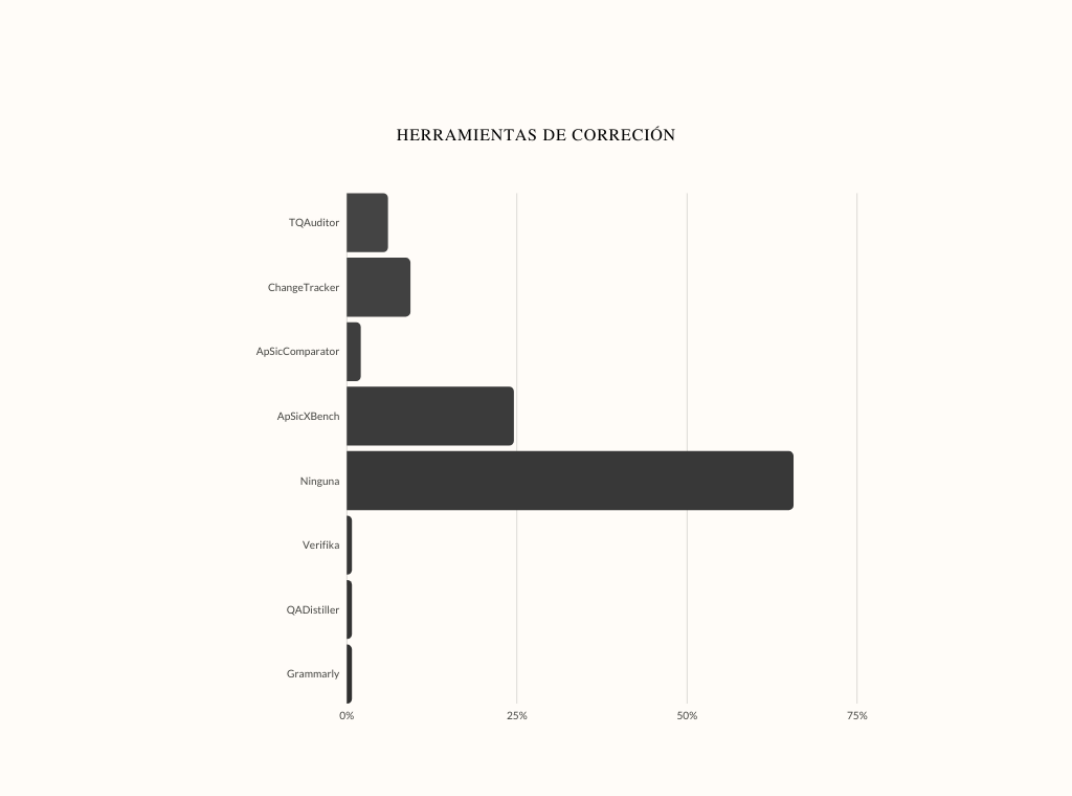

Figura 28 – Gráficos herramientas corrección

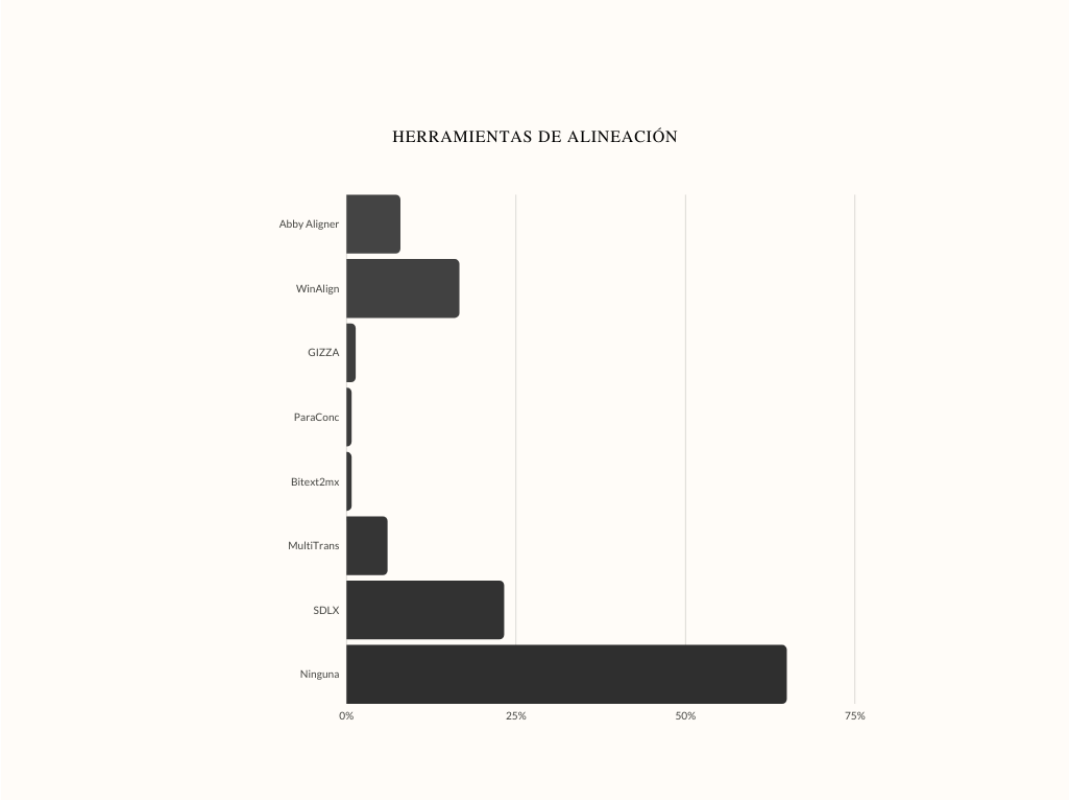

Figura 29 – Gráficos herramientas alineación

8 Intervención Humana

8.1 Traducción Automática en el contexto académico

Los encuestados consideran que se encuentran familiarizados con las herramientas de TA, obteniendo para los estudiantes una valoración de 4.85 y para los graduados la valoración de 5.37, siendo en ambos casos Alta. Se encuentra un aumento significativo entre ambos grupos, que permite afirmar que a medida que el traductor se adentra en el ámbito profesional, asciende la valoración de los conocimientos informáticos respecto a las herramientas utilizadas.

El grupo de egresados encuestado constata mediante una valoración de 2.94 (Baja) que no estuvo en contacto con herramientas de traducción automática en su etapa de formación académica. En cambio, los estudiantes afirman con una valoración de 4.44 (Media), que en la actualidad se encuentran en interacción con herramientas de TA durante la etapa formativa. Ambos grupos concuerdan, que la capacitación en este tipo de herramientas

es esencial y debe formar parte del plan de estudios obligatorio en una carrera de traducción. Los egresados dan una valoración de 5.83 en este punto, mientras que los estudiantes lo hacen en 6.27, ambas valoraciones son altas.

De igual forma ambos grupos consideran necesario dominar estas herramientas, para ambos casos se obtuvo una valoración alta, siendo 6.19 para egresados y 6.28 para estudiantes.

Se puede constatar que, en la etapa formativa actual, si bien los estudiantes están en contacto con tecnología adecuada a la traducción, y conocen la necesidad de la misma en el mercado laboral, no se logra tener un contacto en profundidad con herramientas tecnológicas de TA.

Como un caso no menor a tener en cuenta, ambos grupos consideran con una valoración media que las herramientas de traducción automática, actualmente se encuentran sobrevaloradas en el mercado, los estudiantes brindan una valoración de 3.84 y los egresados 4.10 respectivamente.

8.2 Eficiencia de las herramientas de TA

Los estudiantes afirman que al momento de traducir utilizan herramientas de TA, ya sean motores de traducción o memorias de traducción. El grupo de encuestados egresados respaldan esta afirmación con una valoración de 4.64, mientras que los estudiantes 4.69, en ambos casos la misma es considerada alta.

Los egresados definen que no participan de la creación de motores de traducción automática de ningún tipo, con una valoración de 2.40 (Baja) constatan el casi nulo acercamiento con los procesos de creación de motores de TA. Para el caso de los estudiantes, la valoración obtenida es de 2.02 siendo considerada muy baja.

Si bien el proceso de creación de un motor consta de muchas fases, entre ellas la creación de corpus alineados para que los motores sean entrenados, se le consulta a los encuestados si alguna vez participaron de la creación de un corpus especifico que luego sería utilizado para el entrenamiento de un motor. Los egresados detallan una valoración baja de 2.40 mientras que con los estudiantes se obtiene una valoración de 1.98 siendo la misma considerada muy baja.

Se puede constatar la poca participación de profesionales de la traducción al momento de crear motores o entrenarlos. De esta manera se puede definir que el traductor promedio,

actualmente es considerado un consumidor de las tecnologías de TA, pero no así un productor de la misma. Si bien se nota un leve incremento en la valoración a medida que el traductor tiene años de experiencia en la profesión, los niveles son realmente bajos.

Ambos grupos encuestados consideran con una valoración media, que se obtienen resultados positivos con la TA, los egresados brindan una valoración de 4.30 mientras que los estudiantes 4.25 respectivamente.

Si bien ambos grupos consideran con una valoración alta (5.48 para egresados y 5.21 para estudiantes), facilita las tareas del traductor, ambos grupos coinciden que la inteligencia artificial nunca va a ser capaz de tener en cuenta aspectos culturales al momento de realizar una traducción. Ambos grupos los confirman con una valoración de 3.14 (baja).

8.3 Herramientas utilizadas

8.3.1 Traducción Asistida por Ordenador

Solo el 0.7% del universo encuestado, utilizó alguna vez alguna herramienta OpenSource para crear o entrenar un motor de traducción automática ya sea estadístico o neuronal. El 79.5% utiliza algún software para crear memorias de traducción, ya sea SDL Trados Studio, Wordfast, MemoQ o Virtaal. El 80% del universo encuestado utiliza motores neuronales como DeepL o Google Translate, pero solo como usuarios ya que estas plataformas son privadas y no permiten ser entrenadas ni parametrizadas. De igual manera el 21.1% utiliza el motor estadístico de Microsoft en su versión gratuita. El 3.3% de los encuestados afirma que no utiliza ningún tipo de herramienta TA para sus traducciones.

8.3.2 Herramientas de corrección

El 65.6% de los encuestados no utiliza ningún tipo de herramienta especifica de corrección o pos edición. Mientras que el 24.5% utiliza ApSicXBench, el 9.3% usa changeTracker, y el 6% TQAuditor.

8.3.3 Herramientas de alineación

Estas herramientas se utilizan para crear corpus específicos para que sean utilizados para entrenar motores o crear memorias de traducción específicas. El 64.9% detalla que nunca utilizó ninguna herramienta de este tipo, mientras que el 23.2% usa SDLX, el 16.6% WinAlign, el 7.9% Abby Aligner y solo el 1.3% usa GIZZA.

8.4 Conclusiones

A modo de resumen se detallan las conclusiones que se extrajeron mediante el análisis de los resultados arrojados por la encuesta.

- La interacción con herramientas de TA en la etapa académica hoy en día es mucho mayor a la de hace unos años, aunque aún no es suficiente.
- Las herramientas utilizadas, en su mayoría son herramientas de creación de memorias de traducción, o traductores con arquitectura SaaS, los cuales solo pueden ser ejecutados como usuarios finales.
- El profesional de hoy en día casi no tiene contacto con la creación o entrenamiento de motores, siendo esta tarea realizada por recursos especializados en áreas tecnológicas. Las herramientas OpenSource que se encuentran disponibles hoy en el mercado, no cuentan con una interfaz amigable, esto quiere decir que el usuario que desea crear, y entrenar un motor debe contar con conocimientos técnicos avanzados. De esta manera es difícil para el Traductor profesional convertirse en un productor de tecnología, siendo su rol enteramente el de consumidor.
- La percepción del traductor se encuentra reticente frente a las tecnologías de traducción, particularmente con la inteligencia artifical, donde estiman que va a ser imposible para la misma, obtener una traducción de calidad.

El tipo de dato recogido, puede ser considerado subjetivo, ya que se basa en opiniones. El objetivo de la encuesta es recolectar datos y entender la percepción del profesional traductor.

9 Análisis de datos

9.1 Introducción

En este capítulo se detallan los resultados obtenidos en cada uno de los experimentos, junto con las métricas obtenidas del análisis automático de los resultados.

9.2 Métricas Utilizadas

Mediante la evaluación automática, ejecutada mediante el servicio provisto por la Universidad de Autónoma de Barcelona, es posible obtener la puntuación BLEU de cada una de las traducciones generadas por los motores.

Si bien la métrica BLEU, es la más importante, no es la única que se obtiene mediante el análisis. Las métricas utilizadas en este trabajo son las siguientes:

- **BLEU** (Bilingual Evaluation Understudy), esta métrica creada en 2002 por IBM, es el método más conocido y popular para analizar traducciones generadas por herramientas de TA. Se compara la traducción humana con la obtenida por el motor analizando la precisión y fluidez. Los resultados que se obtienen van de 0 a 100 y se utiliza la siguiente escala para determinar la calidad del resultado.

Tabla 18

Escala BLEU

Puntuación	Interpretación
<10	Obsoleta
10 - 19	Difícil de captar esencia
20 - 29	La esencia es clara pero tiene errores significativos
30 - 40	Comprensible por buenas traducciones
40 - 50	Alta calidad
50 - 60	Muy alta y fluida
>60	Calidad excepcional

Nota: Esta tabla muestra la escala publicada por el servicio Google Cloud Translation

WER (Word Error Rate), esta métrica detalla porcentualmente la cantidad de palabras que se eliminan, se sustituyen o agregan para que el texto traducido se parezca al que se utiliza como referencia.

9.3 Traducción croata > inglés

Tabla 19

Listado de oraciones traducidas hr>en

Apertium	Joshua	OpenNMT
Housing near the school	Lives near the school.	He lives near school
That is between our	That's between us.	This is between us.
From Sunday no rain	No rain since Sunday.	There's been no rain since Sunday.
Passed past his old house.	He walked past his old house.	He passed by his old house
Tramvaj not drives because of problema power flow	Tram is not driving due to power failure.	The tram does not run due to power outages
House is beside school	House near the school.	The house is next to the school
Workers got payment despite crisis	Workers got a cry despite the crisis.	Workers get their salaries in spite of the crisis

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 1

Tabla 20

Listado de frases traducidas hr>en

Apertium	Joshua	OpenNMT
I sing now	I sing now	I'm singing now.
Constantly read	I read all the time	I keep reading
Read novelty	I read the papers	I read the paper
Does fall rain?	Is it raining?	Is it raining?
That book is for phase.	That book is for me.	That book is for me
Bird flies over house.	The bird flies hope home	Bird flying hope house
Sense that dont have air.	I feel like I miss the air	I feel like I'm out of air

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 2

Tabla 21

Texto traducido hr>en

Apertium	Joshua	OpenNMT
Not ask me noćas nothing empty me that šutim	Don't ask me tonight anything let me keep my mouth shut	Don't ask me tonight. Don't let me shut up
Ja noćas should peace	I tonight I need peace	I need peace tonight
Old hurt again cook my bitke farther flow, dušo	Old wounds again burn my battles still flowing, honey	The old wounds are again baking my battles, baby
Those nemaš nothing with those	You don't have anything to do with it	You have nothing to do with it
With your source my se duša napila	With your sources my soul got drunk	From your source, my soul got drunk
Žedna your years	Thirsty your age	Thirsting your age
And now mamurna asks where is comfort	And now hung over asking where is the comfort	And now the hangover asks where the comfort is
Where is youth missing	Where has youth gone	Where the youth disappeared
Go gave ja ugh follow, sometimes to tebe svratim	Go the days I'm following them, sometimes until you drop	The days go I follow them, sometimes I come to you
Dušo search zaborav	Baby I'm looking for oblivion	Honey, I'm looking for oblivion
Molim hours that se returned clues her walk	Please hours to return traces her walk	I ask hours to return to the traces of her walk
Silent as that is here	Quiet like that	Quiet as it is
All also smells on nju, and gave, and morning which will came	Everything still smells like her, and the day, and the morning will come	Everything still smells like her, and day, and morning you will come
After this night, night without dream	After this night, the night without sleep	After this night, night without sleep
And dvjesto year that ugh count in loneliness	And two hundred years that counting them alone	And two hundred years to count them in solitude
Since left	Since she's gone	Since she left

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 3

9.3.1 Resultados

Tabla 22

Métricas hr>en

Motor	BLEU	WER
Apertium	15.25	53.27
Joshua	33.65	37.38
OpenNMT	55.11	22.43
Bing	38.11	40.19
DeepL	27.68	48.60

Nota: Esta tabla muestra las métricas obtenidas con cada motor utilizado para el par de idiomas croata > inglés

En el par croata > inglés, el puntaje más alto lo obtuvo OpenNMT, en tanto Apertium obtuvo el puntaje más bajo, de esta manera y con los resultados obtenidos se encuentra que particularmente, este motor tuvo muchas dificultades en la traducción del texto completo, podemos confirmar que la arquitectura basada en reglas no brinda buenos resultados con textos de índole literario. También se detallan los resultados obtenidos con los motores gratuitos Bing y DeepL, en este escenario en particular se ve un mejor resultado utilizando Bing como motor estadístico, podemos confirmar que para un par de lenguas no tan comúnmente utilizado como el croata > ingles, los servicios gratuitos basados en RNA no brindan una traducción considerada de calidad.

Probablemente debido a que en comparación con el par español > inglés, el par croata > inglés está entrenado con corpus menos extensos y con menor riqueza.

9.4 Traducción español > inglés

Tabla 23

Listado de oraciones traducidas sp>en

Apertium	Joshua	OpenNMT
It lives near of the school.	Lives near the school	He lives near the school
This is between us.	This is between us.	This is between us.
The Sunday will not rain.	Sunday will not rain.	It won't rain on Sunday.
It happened at the side of his ancient house.	He passed by his old house	Passed by his old house
The tram does not work by fault of electricity.	The tram does not work due to lack of electricity	The tramway does not operate due to lack of electricity.
The house is at the side of the school.	The house is next to the school.	The house is next to the school.
The workers received the payment in spite of the crisis.	Workers received the payment despite the crisis.	The workers were paid despite the crisis.

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 4

Tabla 24

Listado de frases traducidas sp>en

Apertium	Joshua	OpenNMT
I am singing.	I'm singing.	I am singing.
I follow reading.	I keep reading.	I continue reading.
I read the newspaper.	I read the paper.	I read the newspaper.
It is raining?.	Is it raining?	Is it raining?
This book is for me.	That book is for me.	That book is for me.
The bird flies on the house	The bird flies over the house.	The bird flies over the house
Seat that it is missing me the air.	I feel like I'm short of air.	I feel short of breath.

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 5

Tabla 25

Listado de perífrasis verbales traducidas sp>en

Apertium	Joshua	OpenNMT
It has begun to rain.	It's started raining.	It has started to rain.
The sun is about to go.	The sun's about to rise.	The sun is about to rise.
I am reading the newspaper.	I'm reading the paper.	I am reading the newspaper.
I go to go going.	I'm going to go.	I'm going to go.
You have to eat more.	You have to eat more.	You have to eat more.
I go it to you to explain.	I'll explain.	I will explain it to you.
Tomorrow I have to go to Paris.	I have to go to Paris tomorrow.	Tomorrow I have to go to Paris.

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 6

Tabla 26

Listado de expresiones idiomáticas traducidas sp>en

Apertium	Joshua	OpenNMT
Be in good hands	Being in good hands	Being in good hands
Have hangover.	Having a hangover.	Having a hangover.
Be all heard.	To be all ears.	Be all ears.
We go!	Let's go!	Let's go!
Put the leg.	Screw up.	Screwing up.
Work of Sun to Sun	Working from Sun to Sun	Working from Sunrise to Sunset
Be between the sword and the wall.	Being between the sword and the wall.	Being between the sword and the wall.

Nota: Esta tabla muestra los resultados de la traducción de los elementos de la tabla 7

9.4.1 Resultados

Tabla 27

Métricas sp>en

Motor	BLEU	WER
Apertium	36.81	39.71
Joshua	56.39	25.84
OpenNMT	60.29	24.88
Bing	50.44	33.97
DeepL	59.63	26.79

Nota: Esta tabla muestra las métricas obtenidas con cada motor utilizado para el par de idiomas español > inglés

En el par español > inglés se obtuvo la mejor puntuación con OpenNMT, es preciso detallar que todos los motores obtuvieron un puntaje alto, incluido Apertium. Esto principalmente es debido a dos motivos fundamentales, por un lado, no se ejecutaron pruebas sobre textos de índole literario o artística como en el par de lenguas anterior. Es por eso que el motor basado en reglas tuvo un mejor resultado. El segundo motivo es que este par de lenguas, español > inglés, es mucho más compatible que el anterior, debido a que, si bien no tienen la misma raíz idiomática, comparten estructuras. Los servicios gratuitos de Bing y DeepL también obtuvieron resultados considerados de calidad. Siendo DeepL el motor que especificamente obtuvo los mejores resultados con las expresiones idiomáticas. Esto se debe directamente a la calidad de los datos utilizados en el entrenamiento de sus motores.

9.5 Observaciones

Luego de analizados los resultados, se define que la traducción con mayor puntaje se obtuvo a través del motor neuronal. Cabe destacar que los servicios de Bing y DeepL se analizaron para complementar la investigación pero no se tienen en cuenta a la hora de analizar el resultado general. Estos servicios utilizados bajo el concepto de caja negra, donde no es posible manipular ni parametrizar sus procesos de entrenamiento.

10 Conclusiones

La contribución principal de este trabajo es brindar un análisis y comparación empírica de los tres motores de TA definidos en el alcance de la investigación. Adicionalmente, se profundiza en el rol del profesional traductor y su participación dentro de la generación de tecnología aplicada a la traducción.

En base a los resultados arrojados en la experimentación se concluye que las redes neuronales artificiales presentan un rendimiento superior frente a las otras tecnologías analizadas. La capacidad de procesamiento computacional y la obtención de datasets son un factor clave, y a su vez limitante, para el desarrollo y entrenamiento de modelos neuronales de alto rendimiento.

Es una cuestión de tiempo para que la totalidad de motores estadísticos, que se encuentran en el mercado, sean reemplazados por motores neuronales. La evolución y creación de nuevos datasets de entrenamiento avanza a diario.

Con el desarrollo de las tecnologías aplicadas a la traducción se debe entender que el rol del profesional traductor debe avanzar y evolucionar. La ejecución de las pruebas demuestran que por más que se hayan obtenido resultados favorables en la traducción de textos, ningún motor logró brindar una traducción ciento por ciento correcta. El traductor humano es un recurso necesario e imprescindible de la post-edición de los resultados.

El rol humano avanza y la tecnología facilita las tareas consideradas como “mecánicas”. Si bien la IA permite, mediante patrones, detectar aspectos que para una tecnología anterior resultaban imposibles, existen factores culturales y sociales que actualmente solo pueden ser analizados por un traductor humano.

Es oportuno señalar que para lograr desarrollos y avances en el área de la tecnología aplicada a la lingüística, el traductor profesional debe estar capacitado y completamente

preparado para hacer uso de las tecnologías disponibles, tanto en la creación como en el entrenamiento de motores neuronales. Éste es uno de los aspectos donde se observan más falencias. El traductor profesional no está inmerso en las ciencias aplicadas actuales. No es capacitado en sus etapas formativas y se genera un grado de aprehensión considerable.

Gracias a la combinación de tecnologías como el reconocimiento de imagen, reconocimiento de voz e inteligencia artificial, es posible facilitar la comunicación entre humanos que hablan diferentes idiomas de una manera casi inmediata, sin embargo ninguna traducción puede ser considerada de calidad. Cabe destacar que el humano no capacitado se convierte en dependiente de la tecnología y no en usuario productor/generador de la misma.

Tal como se expresó anteriormente, las traducciones no arrojaron un resultado cien por ciento correcto.

La hipótesis planteada no se puede comprobar y no es posible obtener una traducción de calidad totalmente automatizada sin ningún tipo de post-edición humana.

Con los suficientes datasets correspondientes es posible crear traductores para cualquier par de lenguas sin necesidad de usar una lengua intermedia cómo el inglés.

11 Futuras Líneas de Investigación

Como futuras líneas de investigación, se resumen todos los aspectos que determinan ser claves para desarrollar en próximos trabajos científicos. Dichos trabajos deben ser analizados en profundidad, ya que, se encuentran por fuera de los objetivos definidos y planificados en este trabajo. Es necesario llevar a cabo cualquier tipo de investigación basándose en las arquitecturas neuronales, ya que, éstas representan el estado del arte. Como la capacidad de procesamiento computacional presenta un limitante, es necesario adquirir un servicio cloud privado para poder realizar las fases de procesamiento en la nube y así entrenar el par de lenguas hr > en con corpus más extensos.

De esta manera se puede ahondar en la traducción automática de textos literarios o de carácter artísticos, ya que es donde se encontraron la mayor cantidad de falencias en los resultados. Esta evaluación involucra directamente la participación de traductores literarios. Para tener un análisis objetivo, se deben estudiar los resultados en profundidad y no solo con métricas automáticas. Plataformas como Google Colaborative, brindan la posibilidad de desarrollar modelos de aprendizaje automático en ambientes Cloud. De esta manera se puede

analizar la posibilidad de prescindir del procesador físico necesario para realizar tareas de entrenamiento de modelos.

Referencias

Casacuberta Nolla, Francisco. (2017) *Traducción automática neuronal*. Revista Tradumatica, n° 15.

Cires¸an, C, D., Meier, U., Masci, J., Gambardella, M, L., Schmidhuber, J. (2010). *Flexible, High Performance Convolutional Neural Networks for Image Classification*. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 1237-1242.

Dharani Kumar Palan (2018). *Statistical Machine Translation of English Text to API Code Usages: A comparison of Word Map, Contextual Graph Ordering, Phrase-based, and Neural Network Translations*. Concordia University, Quebec, Canada.

Escolano, C (2017). *Generación morfológica con algoritmos de aprendizaje profundo integrada en un sistema de traducción automática estadística*. Procesamiento del Lenguaje Natural, Revista n° 59.

Etchegoyhen, T (2018). *Estimación Automática de Calidad de Traducción Mediante Aprendizaje Automático Supervisado y No-Supervisado*. Procesamiento del Lenguaje Natural, Revista n° 61.

F. J. Och and H. Ney (2000). *Giza++: Training of statistical translation models*.

Fischer, L, Laubli, S (2020). *What's the Difference Between Professional Human and Machine Translation? A Blind Multi-language Study on Domain-specific MT*, Department of Computational Linguistics, University of Zurich.

G. Neubig and T. Watanabe (2016). *Optimization for statistical machine translation: A survey*. Computational Linguistics..

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu (2002). *Bleu: a method for automatic evaluation of machine translation*. In Proceedings of the 40th annual meeting on association for computational linguistics.

A. G. Schuth and C. Monz (2010). *Tuning methods in statistical machine translation*..

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

Lopes, V, Farajian M, Bawden R, Zhang, M , Martins A (2020) *Document-level Neural MT: A Systematic Comparison*. University of Edinburgh, Scotland, UK.

Luong, M., Pham, H., y Manning, C. D. (2015). *Effective approaches to attention-based neural machine translation*.

Oliver, A. (2016). *Herramientas tecnológicas para traductores*. Barcelona: Editorial UOC. ISBN: 978-84-9116-048-9.

Olmedo Ruiz, M (2018) *Los tipos de traducción automática y su evaluación mediante paráfrasis verbales y expresiones idiomáticas (alemán – español)*. (Tesis de Maestría). Universidad Autónoma de Barcelona, España.

Oropeza, A, C. (2007). *Modelado y Simulación de un Sistema de Detección de Intrusos Utilizando Redes Neuronales Recurrentes (tesis de pregrado)*. Universidad de las Américas Puebla, México.

Padilla, Alberto (2012): *El efecto del diseño: Sesgo y estimación varianza*, Working Papers, No. 2012-18, Banco de México, Ciudad de México.

Pierre, J, G., Arteaga, R. (2015). *Aplicación del aprendizaje profundo ("Deep learning") al procesamiento de señales digitales* (tesis de pregrado). Universidad Autónoma de Occidente, Colombia.

Stefaniak K (2020*). Evaluating the usefulness of neural machine translation for the Polish translators in the European Commission*, European Commission, Directorate General for Translation.

Torres Badia, G (2017). *Traducción Automática Interactiva Basada en Segmentos de Palabras* (Tesis de Maestría) Universidad de Valencia, España.

Velo Fuentes, E. (2020). *Introducción a los métodos Deep Learning basados en Redes Neuronales* (Tesis de maestría) Universidad de Coruña, España.

Villa, J. (2009). *El dominio de la lingüística más allá de las ciencias naturales y exactas*. Universidad Nacional de Monterrey, México.

Wulliamoz, B, (2018). *Percepción del traductor frente a la calidad de la traducción automática neuronal y sus diferencias con la humana* (Tesis de grado). Universidad Católica de Valparaíso, Chile.

Zhan, Y., Vogel, S. & Waibel, A. (2017) *Interpreting BLEU/NIST scores: ¿How much improvement do we need to have a better system?*

Lloret H, Suarez Cueto, A (2021) *Generación del Lenguaje Natural: retos y desafíos científicos*. Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos, España.

Gelbukh, A, (2010). *Procesamiento de Lenguaje Natural y sus Aplicaciones*. Komputer Sapiens Año II, Vol. I.

Delisle, J (2003). *Íkala, revista de lenguaje y cultura*, vol. 8, núm. 14. Universidad de Antioquia, Colombia.

Viver Sorolla, P, (2018). *La evaluación de las herramientas de traducción automática (TA) desde la perspectiva del traductor: Google Translate, Bing, Babylon y Systran*. Universidad de Vallalodid, España.