UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Traducción interactiva basada en segmentos usando grandes modelos de lenguaje

TRABAJO FIN DE MÁSTER

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Autor: Sergio Gómez González
Tutor: Francisco Casacuberta Nolla
Cotutor: Miguel Domingo Ballester

Curso 2023-2024

Resum

Hui dia les màquines no són capaces d'oferir traduccions de suficient qualitat en molts àmbits per si mateixes. No obstant això, el temps que empren a obtindre estes traduccions és molt de menor que el que necessitaria un traductor humà. La traducció automàtica interactiva sorgix com un paradigma que combina la iniciativa de la màquina i la supervisió d'un humà expert. En concret, en un sistema de traducció interactiva basat en segments el procediment és el següent: primer, l'ordinador oferix una traducció preliminar, després, el traductor humà verifica quines parts de la traducció són correctes i introduïx alguns (pocs) canvis. Amb la nova informació proveïda per l'humà, l'ordinador ha de ser capaç d'elaborar una traducció de millor qualitat. Després de diverses iteracions d'este procediment s'aconseguirà una traducció de bona qualitat (a causa de la supervisió humana) amb un esforç menor del qual empraria un traductor humà per si mateix (gràcies al suport de la màquina).

L'auge dels grans models de llenguatge i les possibilitats que ens oferixen conviden a introduir-los també en este àrea. En este treball s'han seleccionat els models que s'han estimat més apropiats per a utilitzar-los en un sistema de traducció interactiva basat en segments i en prefixos. Addicionalment s'ha realitzat un *fine tuning* dels models per a adaptar-los a la tasca i direcció de traducció adequades.

Per a avaluar les prestacions del sistema a desenvolupar s'utilitzarà una porció dels *datasets* Europarl i HPLT. Concretament s'avaluaran les traduccions entre els parells de llengües francés-anglés, alemany-anglés, espanyol-anglés, gallec-anglés i suajili-anglés. Per a cada direcció de traducció es realitzarà un *fine tuning* de cada model, s'adaptarà la generació de text a una generació en feix restringida i es realitzarà una simulació d'interacció amb un traductor humà. Com a mètriques d'avaluació s'usaran tant la puntuació BLEU i TER per a avaluar la qualitat inicial del sistema, com la taxa d'error de paraula (WSR) i la taxa d'accions de ratolí (MAR) de la simulació.

Paraules clau: Model de llenguatge, traducció, traducció interactiva, traducció interactiva basada en segments, intel·ligència artificial, aprenentatge automàtic, generació restringida

---

Resumen

Hoy en día las máquinas no son capaces de ofrecer traducciones de suficiente calidad en muchos ámbitos por sí mismas. Sin embargo, el tiempo que emplean en obtener dichas traducciones es mucho menor que el que necesitaría un traductor humano. La traducción automática interactiva surge como un paradigma que combina la iniciativa de la máquina y la supervisión de un humano experto. En concreto, en un sistema de traducción interactiva basado en segmentos el procedimiento es el siguiente: primero, el ordenador ofrece una traducción preliminar, después, el traductor humano verifica qué partes de la traducción son correctas e introduce algunos (pocos) cambios. Con la nueva información provista por el humano, el ordenador debe ser capaz de elaborar una traducción de mejor calidad. Tras varias iteraciones de este procedimiento se conseguirá una traducción de buena calidad (debido a la supervisión humana) con un esfuerzo menor del que emplearía un traductor humano por sí mismo (gracias al apoyo de la máquina).

El auge de los grandes modelos de lenguaje y las posibilidades que nos ofrecen invitan a introducirlos también en este área. En este trabajo se han seleccionado los modelos que se han estimado más apropiados para utilizarlos en un sistema de traducción interactiva basado en segmentos y en prefijos. Adicionalmente se ha realizado un *fine tuning* de los modelos para adaptarlos a la tarea y dirección de traducción adecuadas.

Para evaluar las prestaciones del sistema a desarrollar se utilizará una porción de los *datasets* Europarl y HPLT. Concretamente se evaluarán las traducciones entre los pares de lenguas francés-inglés, alemán-inglés, español-inglés, gallego-inglés y suajili-inglés. Para cada dirección de traducción se realizará un *fine tuning* de cada modelo, se adaptará la generación de texto a una generación en haz restringida y se realizará una simulación de interacción con un traductor humano. Como métricas de evaluación se usarán tanto la puntuación BLEU y TER para evaluar la calidad inicial del sistema, como la tasa de error de palabra (WSR) y la tasa de acciones de ratón (MAR) de la simulación.

Palabras clave: Modelo de lenguaje, traducción, traducción interactiva, traducción interactiva basada en segmentos, inteligencia artificial, aprendizaje automático, generación restringida

Abstract

Nowadays machines are not capable of offering translations of sufficient quality in many applications. However, the time they spend obtaining these translations is much less than what a human translator would need. Interactive machine translation emerges as a paradigm that combines the initiative of the machine and the supervision of an expert human. Specifically, in a segment-based interactive translation system the procedure is as follows: first, the computer offers a preliminary translation, then the human translator verifies which parts of the translation are correct and introduces some (few) changes. With the new information provided by the human, the computer should be able to produce a better quality translation. After several iterations of this procedure, a good quality translation will be achieved (due to human supervision) with less effort than a human translator would use by himself (thanks to the support of the machine).

The rise of large language models and the possibilities they offer invite us to also introduce them in this area. In this work, the models that may be most appropriate for use in an interactive translation system based on segments and prefixes have been selected. Additionally, we have fine tuned the models to adapt them to the right task and translation direction.

To evaluate the performance of the system to be developed, a portion of the Europarl and HPLT datasets will be used. Specifically, translations between the French-English, German-English, Spanish-English, Galician-English and Swahili-English language pairs will be evaluated. For each translation direction, a fine-tuning of each model will be carried out, the text generation will be adapted to a restricted beam generation and a simulation of interaction with a human translator will be carried out. Both the BLEU and TER, as an estimation of the intial quality of the system, as well as the word error rate (WSR) and the mouse movement rate (MAR) of the simulation will be used as evaluation metrics.

Key words: Language model, translation, interactive translation, segment based interactive translation, artificial inteligence, machine learning,restricted generation

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

La llegada de los grandes modelos de lenguaje (o *LLM* por sus siglas en inglés) preentrenados ha supuesto un gran impulso en una amplia variedad de tareas de aprendizaje automático. A día de hoy parece que utilizar estos modelos como punto de partida nos ofrece mejores resultados que empezar con un modelo desde cero. Además, hay algunas tareas para las que ni siquiera hace falta reajustar estos modelos para obtener buenos resultados. Es por ello que hemos decidido aplicar LLMs a la tarea de traducción. Además, se usarán con un esquema de traducción interactiva para ofrecer apoyo a un traductor humano y potenciar tanto la velocidad de traducción como el esfuerzo requerido para obtenerla. Este esquema se basa en iteraciones en las que el sistema ofrece proposiciones de traducción que el humano ha de validar, o corregir ciertas palabras. La información introducida por el ser humano es tenida en cuenta para ofrecer la siguiente proposición de traducción sobre la misma oración origen. De esta forma trataremos de comparar los resultados ofrecidos por los distintos LLM que se van a analizar.

1.1 Motivación

El presente trabajo se centra en el area de la traducción automática de textos. En este ámbito es cierto que todavía no podemos usar un esquema de traducción completamente automático y asegurar la calidad de la traducción [41]. No obstante la incorporación de componentes auxiliares al traductor humano ha sido siempre una forma de aumentar su rendimiento. No sería desacertado apuntar a los diccionarios bilingües como las primeras formas de apoyo al traductor. Con la llegada de los ordenadores hemos ido mejorando las herramientas de apoyo hasta llegar al estado actual del arte, donde la tarea del traductor es mayormente de verificación de resultados. No obstante, aun hoy se sigue necesitando la intervención humana para poder obtener traducciones de alta calidad. Por tanto, las herramientas más usadas siguen un marco de traducción interactiva o posedición. Estos sistemas, en los que el ser humano se sirve de herramientas automáticas, ofrecen el mejor rendimiento con la mayor calidad de traducción. Es por ello que este trabajo pretende apoyar a la investigación en dicho campo.

Además, en los últimos tiempos la llegada de los grandes modelos de lenguaje ha supuesto un nuevo avance en los sistemas de traducción. Si bien es cierto que los LLM suelen requerir de numerosos recursos también es verdad que suelen elaborar mejores traducciones. Así es que nos gustaría evaluar distintos grandes modelos de lenguaje en la tarea a la que se dedica este trabajo: la traducción automática interactiva.

1.2 Objetivos

En este proyecto se pretende implementar un sistema de traducción interactiva basado en grandes modelos de lenguaje. Hablamos de un sistema en el que el humano supervisa las traducciones obtenidas por el ordenador. De esta forma el traductor humano señalaría las partes de la traducción erróneas y la máquina ofrecería una traducción en la que, a priori, se corregirían esos fallos. Para ello, se ajustarán varios modelos preentrenados con el *dataset* Europarl y el HPLT. Este mismo *dataset*, o más concretamente su partición de test, será la referencia en base a la cual comparemos las prestaciones ofrecidas por cada modelo de lenguaje. Concretamente, buscamos comparar el esfuerzo que requiere el sistema para obtener una traducción de alta calidad.

Nos gustaría destacar que se van a utilizar varios modelos de lenguaje para estudiar las prestaciones ofrecidas por cada uno. En este sentido exploraremos modelos con arquitecturas y entrenamientos diferentes que les han hecho destacar en algún ámbito. Todos sin distinción han sido preentrenados en varios idiomas, de forma que son intrínsecamente multilingües.

1.3 Estructura de la memoria

A lo largo de este documento aclararemos varios puntos antes de relatar el trabajo realizado y los resultados obtenidos. Así, realizaremos sendas discusiones sobre el estado actual del arte, el sistema de traducción interactiva con grandes modelos de lenguaje, el marco experimental seguido, los resultado obtenidos de los experimentos y, finalmente, las conclusiones.

Primero, en el capítulo 2, nos situaremos en el estado del arte actual de la traducción automática en general y los modelos de lenguaje. Además, profundizaremos en el área de la aplicación de los grandes modelos de lenguaje a la traducción automática y, finalmente, nos centraremos en la traducción automática interactiva.

Después, explicaremos cuál es el sistema de traducción automática interactiva que hemos usado en el presente trabajo en el capítulo 3. Existen múltiples variantes que se explicarán en secciones posteriores y se profundizará más en la traducción interactiva basada en segmentos. También se explicará cómo se aplican los grandes modelos de lenguaje a esta tarea.

Más adelante, en el capítulo 4, expondremos el marco experimental que hemos utilizado para realizar las pruebas de cada uno de los grandes modelos de lenguaje involucrados. Daremos, también, algunos apuntes a fin de conocer el corpus de datos sobre el que hemos trabajado, así como las métricas de evaluación que hemos usado para evaluar los modelos. Después daremos algunos detalles relevantes de los grandes modelos de lenguaje, su arquitectura y su entrenamiento original. Terminaremos el capítulo exponiendo cómo se ha implementado el sistema de traducción interactiva basada en segmentos.

En el capítulo 5, se tratará el análisis de los resultados arrojados por los experimentos. También compararemos los resultados obtenidos en este estudio con otros del ámbito de la traducción automática interactiva.

Finalmente, terminaremos reuniendo las conclusiones deducidas del trabajo realizado y dando algunas ideas de mejora y extensión del presente trabajo en el capítulo 6.

CAPÍTULO 2
Estado del arte

En este trabajo tratamos de utilizar las herramientas de los modelos de lenguaje para la tarea de traducción automática interactiva (IMT por sus siglas en inglés). Así, creemos que realizar un análisis del estado del arte para la herramienta y otro para la tarea puede dar una mejor intuición de la forma en la que ambas han evolucionado y cómo se relacionan hoy en día.

2.1 Traducción automática

Los inicios de la traducción automática se corresponden con la llegada de los algoritmos basados en diccionarios y reglas. Una de las herramientas más veteranas que usaba dichas técnicas es *Systran*[40]. Más tarde, se comprobó que los métodos basados en datos obtenían mejores resultados en tareas reales. Así, los métodos estadísticos se establecieron como dominantes durante las décadas de los noventa y los dos mil. Ejemplos de herramientas que empleaban estos algoritmos son los modelos de IBM[4]. Sin embargo, con la gran ola de las redes neuronales estos algoritmos se vieron sobrepasados y la traducción automática alcanzó un nuevo nivel. Hoy en día la traducción neuronal sigue siendo el paradigma dominante y en el que se basan todas las herramientas competitivas de traducción automática.

Entre las herramientas de traducción de texto más conocidas y utilizadas se encuentran el traductor de *Google*[1], *Amazon translate*[2] o el traductor de *Microsoft*[3]. Las tres son *software* propietario, pero también hay modelos libres que pueden modificarse y adaptarse con posibles ideas de mejora. Tal es el caso del proyecto *OpenNMT*[23] o *Fairseq*[30], que ofrecen una API más sencilla para entrenar nuevos modelos de traducción. Con dichas herramientas no es necesario tener un alto nivel de especialización ni conocimientos profundos de aprendizaje automático. Ofrecen un marco muy atractivo para el prototipado de productos o la iniciación de principiantes en la materia. Si uno sí tiene los conocimientos necesarios puede elaborar sus propios modelos de traducción utilizando los *frameworks* de *pytorch*[4] o *tensorflow/keras*[5] desde cero, al igual que con cualquier otro modelo neuronal.

Por otra parte, también es posible acceder a modelos ya entrenados para traducción. Este es el caso de *MarianNMT*[21], que ofrece modelos para distintas direcciones de traducción. Son modelos de traducción especializados en ciertos pares de lenguas, lo que

[1] https://translate.google.es/?hl=es&sl=es&tl=ca&op=translate
[2] https://docs.aws.amazon.com/translate/latest/dg/what-is.html
[3] https://www.microsoft.com/es-es/translator/
[4] https://pytorch.org/
[5] https://keras.io/

permite obtener buenas traducciones con un pequeño número de parámetros. Sin embargo, ello supone una limitación a la hora de elaborar un sistema de traducción general. En esa línea se han tratado de entrenar modelos que, manteniendo la calidad de traducción, puedan escalar a una cantidad mayor de idiomas. De esta forma, los grandes modelos de lenguaje supusieron el siguiente paso en la evolución de los sistemas de traducción automática.

Hasta ahora nos hemos referido a herramientas de traducción cuya entrada y salida es únicamente texto. No obstante hay sistemas que permiten transcribir el audio directamente a otro idioma. Tal es una de las capacidades de *Whisper*[35], el modelo de transcripción de *OpenAi*[6]. Tratando de explotar al máximo el paradigma, existen modelos que, eliminando un paso, son capaces de traducir de audio a voz, como *Seamless*[7]. En este sentido, los grandes modelos de lenguaje también han impulsado el desarrollo de la traducción automática.

2.2 Grandes modelos de lenguaje

El panorama actual de los modelos de lenguaje se ve dibujado por modelos de tipo *Transformer* [43] de gran tamaño. Desde su proposición en 2017 estos modelos han ofrecido los mejores resultados. Han supuesto una forma de escalar el tamaño de los modelos sin los problemas que ofrecían las redes recurrentes, como las *Long-Short Term Memory* [16]. Así, en los nuevos modelos de lenguaje se ha aumentado el número de parámetros. Este aumento del tamaño del modelo, por sí solo, no permite a los modelos mejorar sus capacidades. A este respecto, como se aclaró con el modelo *Chinchilla*[17], es necesario aumentar de forma similar el tamaño del conjunto de datos de entrenamiento.

Los grandes modelos de lenguaje de hoy en día han superado los tamaños que pueden manejar los ordenadores convencionales. Estos modelos solo pueden ser entrenados desde el principio usando numerosas *GPUs* y vastos conjuntos de entrenamiento. En estas circunstancias dicha opción queda descartada para una gran parte de los profesionales de este sector. Este tipo de proyectos solo pueden ser llevados a cabo por grandes organizaciones. Tal es la razón de ser de los conocidos como modelos preentrenados. Se trata de grandes modelos de lenguaje entrenados con amplios conjuntos de entrenamiento en tareas que son simples pretextos para que el modelo domine la estructura del lenguaje. De esta forma incluso llegan a captar ciertos niveles de comprensión.

Actualmente no es necesario tener conocimientos específicos para benerficiarse de las capacidades de estos modelos. Abundan las aplicaciones en las que se puede interactuar con ellos con una interfaz de *chat* sencilla. Así, el uso más extendido entre la población es el uso de estos modelos mediante *prompts*, i.e. la cadena de texto introducida como consulta en las aludidas aplicaciones. Sin embargo, se necesita someter a los modelos a un segundo entrenamiento para que sean útiles a la hora de funcionar en un entorno "conversacional". Así, se favorecen las respuestas que los usuarios consideran mejores o más útiles. Para tal fin el paradigma de aprendizaje por refuerzo se introduce de forma tan natural como efectiva, tal y como se explica en Ouyang et al. [31].

El relatado es el procedimiento más comúnmente seguido a la hora de ofrecer un gran modelo de lenguaje al público en su sentido más amplio. Los modelos así distribuidos suelen ser aludidos con el apelativo *instruct*, en referencia a su facilidad para ofrecer una buena respuesta a instrucciones. No obstante, también suelen liberarse los modelos de lenguaje "puros". Es decir, los que realmente modelan el lenguaje en el sentido clásico del concepto. Este tipo de modelos no están ajustados a la interacción directa con humanos

[6] https://openai.com/

y pueden ofrecer mejores resultados cuando son reentrenados para tareas más concretas. No obstante, es cierto que deben poseerse tanto los conocimientos como las habilidades y recursos necesarios para realizar el citado reajuste. Es por ello que suelen ser los usados en proyectos destinados a crear herramientas para resolver cuestiones más específicas.

Así, en la situación actual, con la necesidad de usar modelos preentrenados, las licencias de uso de estos modelos marcan las reglas del juego. Mientras hay organizaciones que han optado por ocultar los pesos de sus modelos como una ventaja competitiva hay otras que deciden abrirlos para que puedan ser usados y mejorados por el resto de la comunidad. Existe, por supuesto toda la gama intermedia con, por ejemplo, modelos que pueden ser usados en investigación pero no como parte de un producto o servicio. Empresas como OpenAI[7] de Microsoft, creadores del modelo GPT-3[5], utilizado después en el popular ChatGPT[8], han optado finalmente por guardar para sí mismos sus modelos posteriores. Por otro lado, empresas como Meta[9] (antes Facebook) se inclinan por una estrategia en la que ofrecen sus modelos preentrenados al resto de la comunidad.

Aún teniendo acceso a los modelos preentrenados, es necesaria una máquina potente para poder ajustarlos a la tarea final que se quiere abordar. El número de parámetros de los grandes modelos de lenguaje actuales se cuentan por miles de millones. Como ejemplos podemos citar *T5*[36] con 11000 millones, *Mixtral*[20] con 47000 millones o *Llama2*[42] con 70000 millones. Estos modelos se ofrecen con distintos tamaños, de hecho la cantidad de parámetros citada es la propia de los modelos más grandes de esas familias. Debido a el tamaño de estos modelos, en este trabajo nos centraremos en "pequeños"grandes modelos de lenguaje.

Cabe destacar que últimamente están surgiendo algunos modelos con arquitecturas no *Transformer* que están ofreciendo resultados que alcanzan su mismo nivel en ciertas tareas. Son arquitecturas alternativas que desafían el dominio absoluto de los *Transformers*. Por poner un ejemplo podemos señalar el modelo *Mamba*[13], basado en un modelo de espacio de estados.

2.3 Traducción automática con grandes modelos de lenguaje

Los grandes modelos de lenguajes generativos de hoy en día son entrenados en multitud de lenguas. Ello, a priori, los podría hacer aptos para la tarea de traducción automática. De hecho, es común el uso de *prompts* para inducir a un gran modelo de lenguaje a realizar traducciones[48]. Esta aproximación aprovecha el comportamiento emergente de los grandes modelos de lenguaje conocido como aprendizaje contextual o *in context learning* (ICL). Pese a este nombre, es importante destacar que, realmente, no hay aprendizaje. No hay ninguna actualización de parámetros que así lo indique. Lo que ocurre, en realidad, es que mediante el *prompt* con los ejemplos provistos se guía la generación de texto para obtener unos resultados más acordes a lo que se necesita. No obstante, los grandes modelos de lenguaje guiados con esta técnica para realizar traducciones obtienen buenos resultados, tal y como se indica en Zhu et al. [48].

Otra aproximación, más costosa, recurre a ajustar los grandes modelos de lenguaje específicamente para traducir. En este caso, el modelo sí “aprende” puesto que son los propios parámetros los que se ajustan para realizar la tarea para la que se entrenan. Esta es la aproximación utilizada, por ejemplo, en Navarro y Casacuberta [27] o Zan et al. [47]. También es, sin embargo, una aproximación costosa puesto que se necesita reentre-

[7] https://openai.com/
[8] https://chatgpt.com/
[9] https://www.meta.com/

nar el modelo para habituarlo a la tarea. Sin embargo, hay formas de reducir este esfuerzo de computación, por ejemplo, con el algoritmo *LoRa* [18]. Dicho algoritmo aproxima las matrices de parámetros de los modelos con matrices de menor rango. De esta forma, simplemente hay que entrenar dichas matrices más pequeñas y se reduce drásticamente la computación necesaria. Es importante aclarar que, en principio, las dos aproximaciones expuestas no son excluyentes. En realidad cualquier gran modelo de lenguaje admite *prompts* que lo guíen y puede reentrenarse si tenemos acceso a sus parámetros. Podemos encontrar un ejemplo de este ajuste combinado en Alves et al. [2].

Otra forma de aprovechar el "conocimiento" previo de los grandes modelos de lenguaje es realizar un proceso de *knowledge distillation*, presentado por Hinton, Vinyals y Dean [15]. Más explícitamente, se trata de entrenar un modelo de traducción neuronal, no para obtener las traducciones adecuadas, sino para imitar el comportamiento de un LLM. Las ventajas de este procedimiento son varias. Primero, como hemos indicado antes, se aprovecha el preentrenamiento realizado sobre el LLM para llegar al óptimo del entrenamiento con menos esfuerzo. Además, no se necesita tener las traducciones de las oraciones de entrada, puesto que estas las obtendremos con el LLM. Por último, algunos estudios, como Enis y Hopkins [11], muestran que los modelos de traducción pueden mejorar sus resultados con esta técnica. Como resultado, se obtiene un modelo mucho más reducido que un gran modelo de lenguaje con una calidad de traducción similar. No obstante, se debe recordar que es necesario disponer de los recursos para poder realizar inferencia con el LLM y entrenamiento con el modelo de traducción neuronal. Sin embargo, también es cierto que los procesos de inferencia son considerablemente menos costosos que los de entrenamiento.

Si bien es cierto que lo grandes modelos de lenguaje han aportado mejoras notables en el campo de la traducción, también es cierto que presentan ciertas restricciones. La más obvia de ellas es la cantidad de recursos necesaria para su entrenamiento y uso. Como hemos visto, hay formas de reducir esta restricción con *knowledge distillation*, *LoRa* o cuantización de los parámetros de los modelos. Aún con todas las mejoras que introducen estos modelos, siguen teniendo dificultades para obtener traducciones de alta calidad en algunos casos, como se deduce del Congreso sobre Traducción Automática del año pasado (WMT23)[24]. Es posible que las capacidades generalistas de este tipo de modelos de lenguaje les pasen factura a la hora de compararlos con modelos de traducción neuronal específicos. En esta línea, Hendy et al. [14], proponen un análisis entre los paradigmáticos GPT de *OpenAI* y el mejor modelo de la WMT.

Como hemos visto, los LLMs suponen, en general, un avance en las capacidades de la traducción automática. Aún así continúan teniendo problemas cuando lo que se necesitan son traducciones de alta calidad. En este tipo de traducciones no solo se necesita la correspondencia entre el sentido de la oración original y la traducción. Además es necesario que el estilo de escritura se acomode al contexto en el que se realiza la traducción. En esta línea sigue siendo necesaria la supervisión humana y, por lo tanto, la traducción automática interactiva juega un papel imprescindible.

2.4 Traducción automática interactiva

El campo de la ayuda a la traducción no estuvo tan relacionado con la traducción automática en sí misma desde sus inicios. Proviene de la ayuda a la traducción en la que se utilizaban diccionarios de traducciones frecuentes. Podría decirse que la IMT tal como la conocemos hoy en día comenzó con el proyecto TransType y el sistema propuesto por Langlais, Foster y Lapalme [26]. En dicho artículo se plantea que las proposiciones de la máquina se produzcan en forma de posibles formas de completar la palabra que se está

escribiendo. Esta intuición se ha extendido con las posibilidades de la traducción automática neuronal para poder completar oraciones enteras basándose en el prefijo introducido. Aún con la aproximación basada en ofrecer sufijos de palabras se comprobó que podían obtenerse traducciones “pulsando menos del 40 % de las letras de una traducción"[26]. Así, en el contexto de la traducción automática con métodos estadísticos surgieron más herramientas de IMT. Incluso la Unión Europea decidió invertir en un proyecto de estas características conocido como CasMaCat [1].

Hoy en día los sistemas de traducción interactiva punteros utilizan redes neuronales artificiales. Todo buen sistema de traducción interactiva produce traducciones relativamente válidas como primeras hipótesis. El encargado de asegurar un buen nivel de calidad es el ser humano que interactúa con la máquina. Sin embargo, los sistemas hasta ahora descritos solo proveen ayuda basada en prefijos establecidos de “izquierda a derecha”. Es decir, son sistemas en los que la única información que se le aporta al sistema debe estar localizada en la parte inicial de la traducción. Así, en estos sistemas, únicamente se valida la parte de la oración comprendida entre su inicio y la primera palabra errónea. Esta puede no ser una buena forma de trabajo para algunos traductores humanos o pares de lenguas. Los sistemas actuales son más flexibles, pudiendo proveer al mismo tiempo de ayuda a nivel de palabra y oración, sin estar restringidos a un sentido de traducción fijo como *TranSmart* [19]. Dicho sistema también es capaz de adaptarse al usuario utilizando memorias de traducción para, por ejemplo, no cometer repetidamente el mismo error.

Habitualmente las herramientas software que permiten usar traducción interactiva, también sirven para realizar posedición de todo tipo. Es decir, suelen integrar el mayor número de opciones de trabajo posible para favorecer la versatilidad de su producto. Este es el caso de herramientas como *Synslator* [44], que integra en una sola aplicación ambas posibilidades. También permite sugerencias de traducción a nivel de palabra y de frase. Así, en el panorama actual, las herramientas maximizan su flexibilidad para permitir al traductor trabajar de la forma más cómoda posible.

Sin embargo, este trabajo se ubica en el entorno de la Universidad Politécnica de Valencia y el centro de investigación PRHLT. En este contexto, durante los últimos tiempos ha habido un gran esfuerzo de investigación en las posibilidades que plantea la traducción interactiva basada en segmentos. En artículos como Domingo, Peris y Casacuberta [10], se propone un sistema de ayuda a la traducción basado en la validación de múltiples segmentos de palabras. Sin embargo, en ese momento todavía se utilizan métodos de traducción automática no neuronal. No obstante, en trabajos como Peris, Domingo y Casacuberta [33] o Navarro, Domingo y Casacuberta [28] se usan modelos de traducción neurona precisamente para traducción interactiva basada en segmentos.

CAPÍTULO 3

Sistema de traducción interactiva mediante grandes modelos de lenguaje

El sistema utilizado en este trabajo se basa en el esquema tradicional de la traducción neuronal. En dicho esquema se trata de modelizar la probabilidad de que una traducción (y) se corresponda con cierta oración de entrada (x). Es decir:

$$P(y|x) = \prod_{i=1}^{|y|} P(y_i|y_1^{i-1}, x) \tag{3.1}$$

Donde y_t es el token de la traducción correspondiente a la posición t, e y_s^e es la cadena de tokens desde la posición s hasta la e. De esta forma, la inferencia toma la forma de una búsqueda en el espacio de posibles traducciones que maximice dicha probabilidad. Así, tratamos de maximizar la probabilidad de dicha función:

$$\hat{y} = \arg\max_{y} P(y|x) \tag{3.2}$$

No obstante, el espacio de búsqueda es demasiado extenso como para poder explorarlo por completo. Es por ello, que ni con un modelo de traducción perfecto podríamos obtener las traducciones que fuesen más probables de acuerdo al modelo y, por tanto, ideales. Por ello, la traducción interactiva busca reducir el espacio de búsqueda acotando traducciones que son imposibles. Ello se lleva a cabo ilustrando al modelo con traducciones parciales de la oración original. En esta línea, la traducción interactiva basada en prefijos (como la propuesta de Navarro Martínez [29]) aporta un segmento que se extiende desde el inicio de la traducción. Es decir, la inferencia cambia para encontrar, simplemente, la mejor continuación a dicho prefijo:

$$\hat{y} = \arg\max_{y_p^{|t|}} \frac{P(y_p^{|y|}, y_1^p|x)}{P(y_1^p|x)} = \arg\max_{y_p^{|y|}} P(y_p^{|y_p|}, y_1^p|x) \tag{3.3}$$

Sin embargo, puede darse un paso más y generalizar dicha inferencia para extenderse a un conjunto de segmentos en cualquier posición (F). De esta forma, la labor del modelo de traducción pasa a ser la de rellenar los huecos entre todas las traducciones parciales proporcionadas. Matemáticamente:

$$F = \{f_1, f_2, ..., f_N\} \tag{3.4}$$

$$\hat{y_1}, ..., \hat{y_N} = \arg\max_{y_1,...,y_N} P(f_1, y_1, ..., f_N, y_N|x) \tag{3.5}$$

$$\hat{y} = f_1\hat{y_1}...f_N\hat{y_N} \tag{3.6}$$

En este trabajo hemos tratado de aproximar la probabilidad $P(y|x)$ utilizando grandes modelos de lenguajes multilingües. Sin embargo, estos modelos han sido entrenados para modelar el lenguaje y no como modelos de traducción. Es por ello, que se ha realizado un ajuste de pesos para, aprovechando ese conocimiento previo, adaptarlos a la tarea de traducción.

3.1 Descripción del protocolo de interacción usuario-máquina

El eje central de la traducción interactiva es la colaboración entre el usuario humano y la máquina. En este sentido es vital especificar cómo se lleva a cabo dicha interacción. Usando el esquema de traducción interactiva basada en segmentos el humano debe poder validar ciertos fragmentos de las traducciones ofrecidas. Sin embargo, la validación de ciertos segmentos no es la única acción que el usuario puede realizar. De esta forma, el funcionamiento del sistema propuesto sería el siguiente:

1. El sistema procesa la frase en el idioma original y ofrece una primera hipótesis de traducción en el idioma deseado.

2. Si la traducción es correcta, el traductor humano lo indica pulsando el botón correspondiente. El traductor humano también puede indicar que la traducción correcta sería la resultante de unir todos los segmentos ya validados. En cualquiera de estos dos casos se pasa a la siguiente oración a traducir desde el paso 1.

3. El traductor humano señala para validar tantos segmentos de la hipótesis de traducción como quiera. Es decir, marcará como correctas tantas secuencias de palabras como vea conveniente. El sistema tendrá en cuenta los segmentos validados con respecto del orden en el que se encuentran en la hipótesis de traducción. Los segmentos así validados se mantendrán como tales en iteraciones posteriores.

4. El traductor humano puede marcar segmentos validados que debieran ser sucesivos. Es decir, si un segmento validado es continuación de otro, el traductor humano puede indicar al sistema que, realmente, forman parte de un solo segmento. También es posible fijar un segmento al inicio de la traducción, como un prefijo.

5. El traductor humano puede introducir por teclado una palabra (o más) para guiar al sistema en la traducción que debe ofrecer.

6. El ordenador proveerá de otra traducción teniendo en cuenta toda la información proporcionada por el traductor humano. Se realizará otra iteración del bucle desde el paso 2.

En realidad, los pasos 3, 4 y 5 pueden llevarse a cabo en cualquier orden. Lo realmente importante es que, primero, el sistema ofrece una traducción, después, el traductor humano da una realimentación de las partes correctas de la traducción y, por último, el sistema obtiene información para dar una mejor traducción. Se realizarán tantas iteraciones como sean necesarias hasta que el traductor humano estime que la traducción tiene la calidad suficiente.

El protocolo del sistema basado en prefijos que hemos implementado es el siguiente:

1. El sistema procesa la frase en el idioma original y ofrece una primera hipótesis de traducción en el idioma deseado.

Oración origen: Mister cat was sitting in the roof
Traducción de referencia: Estaba el señor Don Gato sentadito en su tejado

Iter. 0	MT	El señor gato esta sentado en un tejado
Iter. 1	Usuario	**Estaba** El señor gato sentado en un tejado
	MT	Estaba el señor gato sentado en su tejado
Iter. 2	Usuario	Estaba el señor **Don** gato sentado en su tejado
	MT	Estaba el señor Don gato sentadito y tranquilito en su tejado
Iter. 3	Usuario	Estaba el señor Don **Gato** sentadito y tranquilito en su tejado
	MT	Estaba el señor Don Gato sentadito y tranquilito en su tejado
Iter. 4	Usuario	Estaba el señor Don Gato sentadito y tranquilito en su tejado
	MT	Estaba el señor Don Gato sentadito en su tejado

Figura 3.1: Ejemplo de funcionamiento de un sistema de traducción interactiva basado en segmentos como el que se utiliza en este trabajo. En cada iteración el usuario supervisa la hipótesis ofrecida por el sistema validando ciertas partes y corrigiendo otras Los segmentos validados por el usuario se muestran dentro de recuadros y las correcciones introducidas, en negrita.

2. El traductor humano valida todas las palabras correctas desde el inicio de la traducción. En realidad, esta acción sería la misma que para validar un solo segmento. Si toda la traducción es correcta este paso puede obviarse.

3. El traductor humano introduce como corrección la palabra que debería seguir al prefijo validado. Nuevamente, si toda la traducción es correcta este paso no es necesario.

4. Si la corrección es correcta con los cambios introducidos o era correcta sin necesidad de corrección, se indica al sistema pulsando el botón correspondiente. En dicho caso, termina la traducción de esta palabra y comienza nuevamente este proceso con el paso 1 y la siguiente oración. También puede indicarse que la traducción correcta es simplemente el prefijo validado. En caso contrario se pulsa el botón de continuar la traducción.

5. El sistema ofrece una nueva hipótesis que contenga el prefijo validado seguido de la corrección introducida. Se vuelve al paso 2.

Es un protocolo más sencillo que permite validar la traducción únicamente de izquierda a derecha. Su propia elementalidad es su mayor fortaleza. Aunque permita una gama más pequeña de interacciones se centra en utilizar las que pueden ser más efectivas.

En la figura 3.1 puede observarse un ejemplo de funcionamiento de un sistema como el descrito. En ella, a partir de la propuesta inicial simplemente se validan las palabras ”en” y ”tejado”. Además se introduce la palabra ”Estaba” al principio de la traducción. Tras ello, el sistema reelabora la traducción, y esta vez se pueden señalar, además, los segmentos ”el señor” y ”su”. Adicionalmente, se introduce ”Don” como corrección. Con la información reunida, el sistema comete un ”error” introduciendo el segmento ”y tranquilito” que no satisface al traductor. De forma que en la cuarta iteración se indica que los segmentos ”sentadito” y ”en” deben ser sucesivos. Con toda la información proporcionada por el usuario el sistema es capaz de armar una traducción que satisfaga las expectativas de calidad requeridas.

3.2 Aplicación de grandes modelos de lenguaje

A fin de implementar un sistema de traducción interactiva como el descrito se deben seleccionar cuidadosamente los posibles grandes modelos de lenguaje a utilizar. Es vital que estos hayan sido preentrenados en las lenguas entre las que se pretende traducir. En caso contrario tanto el vocabulario como la capacidad de modelar ambos idiomas del propio LLM serían insuficientes. Además, se han seleccionado modelos autorregresivos puesto que generan el lenguaje humano de una forma notoriamente más natural.

Precisamente en la generación de la traducción es donde se encuentra el mayor número de diferencias respecto de otros paradigmas de traducción. Como ya se ha explicado, en el caso que nos ocupa tratamos con una generación restringida, donde la hipótesis de traducción ha de contener determinados segmentos. El algoritmo de búsqueda más generalmente utilizado para obtener hipótesis a partir de modelos generativos es la búsqueda en haz. En este trabajo también ha sido usada para dicho fin con algunas modificaciones para satisfacer las restricciones presentadas al inicio de este capítulo.

Todos los modelos que se han utilizado siguen la arquitectura *Transformer* que tanto éxito ha tenido en el campo de los grandes modelos de lenguaje. Sin embargo, cada uno presenta diferencias, bien en su arquitectura interna o bien en su entrenamiento, que les otorgan cualidades distintas. A este respecto se ha tratado de evaluar las prestaciones de modelos que han apostado por soluciones innovadoras a partir de las cuales han conseguido distinguirse del resto. En el capítulo 4 se relatarán más detalladamente las prestaciones y características de cada uno. Por el momento, destacaremos que hemos utilizado cuatro modelos distintos: *mBART*[39], *M2M*[12], *NLLB*[8] y *Flan-T5*[6].

CAPÍTULO 4
Marco experimental

Es parte de este trabajo el comparar la aplicación de diferentes grandes modelos de lenguaje al sistema descrito en el capítulo anterior. Para ello, se ha definido un marco experimental común que poder aplicar sobre cada modelo. Así, se ha utilizado varios corpus paralelos para, primero, realizar el *fine-tuning* de los modelos y, después, valorar sus prestaciones. Se ha seguido el procedimiento habitual a este respecto: se ha entrenado cada modelo para cada dirección de traducción con la partición de entrenamiento y decidido el mejor *checkpoint* con la de evaluación para, después, analizar sus prestaciones con la partición de test.

4.1 Corpus de datos

Para realizar la experimentación hemos utilizado diferentes corpus en función del par de lenguas entre los que se ha buscado traducir. Primero utilizamos el corpus Europarl[25] como nexo de unión con otros trabajos con el fin de realizar una comparación. Sin embargo, dicho corpus no contenía las lenguas con bajos recursos con las que también teníamos intención de experimentar. Por ello, también se utilizó el corpus de datos del proyecto HPLT (*High Performance Language Technologies*) [9].

4.1.1. Europarl

El corpus de datos para las lenguas con muchos recursos que hemos utilizado en este trabajo fue creado en el año 2001 por Koehn [25] a partir de actas de la Unión Europea (UE). Dichas actas se remontan hasta el año 1996. Se trata de un corpus paralelo que reco-

Tabla 4.1: Número de oraciones paralelas ($|S|$), tokens ($|T|$) y tamaño del vocabulario ($|V|$) en los *datasets* utilizados para cada par de lenguas. Las indicaciones k y M se usan como abreviaturas de los correspondientes múltiplos de 10 según las normas del sistema internacional.

		Europarl						ParaCrawl			
		en-de	de-en	en-es	es-en	en-fr	fr-en	en-gl	gl-en	en-sw	sw-en
Entrenamiento	$\|S\|$	1.9M		2M		2M		956.8k		1.7M	
	$\|T\|$	49.8M	52.3M	57.1M	54.5M	60.5M	54.5M	13.1M	12.5M	21M	20M
	$\|V\|$	15.5k	25.1k	14.7k	25.2k	16k	25.2k	22.5k	28.1k	25.6k	28k
Validación	$\|S\|$	3k		3k		3k		3k		3k	
	$\|T\|$	63.5k	64.8k	78.9k	73k	73.7k	64.8k	40.6k	38.9k	36.3k	34.5k
	$\|V\|$	4.5k	8.8k	4.6k	9.2k	5.3k	8.8k	5.6k	9.4k	5.2k	6.9k
Test	$\|S\|$	2.2k		3k		1.5k		3k		3k	
	$\|T\|$	44.4k	46.8k	70.3k	64.6k	29.5k	26.8k	41.9k	40.3k	36.8k	34.9k
	$\|V\|$	4k	7.4k	4.4k	8.8k	3.5k	5.3k	5.5k	9.4k	5.3k	7.2k

ge oraciones en 11 lenguas oficiales de la UE: danés, alemán, griego, inglés, español, finés, francés, italiano, holandés, portugués y sueco. En su versión de 2005 recogía aproximadamente 30 millones de palabras en cada idioma. En nuestro trabajo solo hemos utilizado las particiones entre francés e inglés, entre alemán e inglés y entre español e inglés. Los tamaños y estadísticas de dichas particiones están disponibles en el cuadro 4.1.

Este corpus fue creado en un contexto de necesidad de datos en el ámbito de la traducción automática usando métodos estadísticos. No obstante, ha sido usado para otros tipos de tareas, tal y como se indica en el artículo de su versión de 2005:

> "Recogimos el corpus Europarl principalmente para asistir en nuestra investigación en traducción estadística interactiva, pero desde que lo hicimos público en su lanzamiento inicial en 2001, ha sido usado para muchos otros problemas de procesamiento de lenguaje natural: desambiguación de palabras, resolución de anáforas, recuperación de información, etc." - *Philipp Koehn*

Pese a que la traducción automática estadística ha sido sobrepasada por los sistemas de traducción neuronal, el corpus Europarl ha trascendido. Sigue usándose para experimentos con sistemas de traducción del estado del arte. Parte de su atractivo es debido a que, al ser actas oficiales de la Unión Europea, el lenguaje utilizado suele ser correcto, sin faltas de ortografía o vulgarismos.

El origen de las oraciones del corpus es la web del parlamento europeo donde están accesibles las actas incluidas. Sin embargo, dichos textos han sido procesados antes de incluirlos en el corpus. Koehn [25] indica que se extrajeron y alinearon *chunks* de texto paralelo que después fueron divididos en las oraciones. Además, se tokenizaron dichas oraciones y, finalmente, se alinearon con sus correspondientes traducciones.

Desde su primera versión en 2001, el corpus ha ido creciendo tanto en tamaño como en idiomas incluidos. La última versión de la que tenemos consciencia es la séptima[1], de mayo del 2012. En ella se incluyen traducciones de procedimientos del Parlamento Europeo de entre 1996 y 2011 en 21 lenguas europeas.

4.1.2. HPLT

El proyecto *High Performance Language Technologies*[2] busca crear grandes corpus de texto en lenguaje natural para potenciar la investigación en procesamiento de lenguaje natural y traducción automática. Además, pretende liberar dichos recursos con licencias de uso abiertas para su utilización masiva. En su primera versión es un corpus anglocéntrico, lo que significa que todas las lenguas cubiertas están conectadas directamente con el inglés. Sin embargo, no contiene oraciones paralelas entre dos lenguas distintas de dicho idioma pívot. Aún así, cubre un total de 18 idiomas distintos y está disponible bajo licencia *creative commons*.

Este corpus no se limita a conectar pares de lenguas en varios idiomas. También dispone de *datasets* monolingües en 75 idiomas distintos. En este trabajo dichos *datasets* no resultan demasiado atractivos, no obstante es una muestra del afán generalista del proyecto HPLT. No en vano pretenden ser un corpus de referencia en toda la disciplina del procesamiento de lenguaje natural. En el artículo de De Gibert et al. [9] se explica cómo fueron limpiados y procesados los textos para los *datasets* monolingües y bilingües. Dichos textos fuente tienen su origen en *Intenet Archive*[3] y *CommonCrawl*[4].

[1] https://www.statmt.org/europarl/.
[2] https://hplt-project.org/
[3] https://archive.org/
[4] https://commoncrawl.org/

Actualmente solo está disponible la primera versión del corpus, publicada en la página web del proyecto OPUS[5] y en la propia del proyecto HPLT. Sin embargo, tal y como indica De Gibert et al. [9] en futuras versiones se pretende dejar de lado el anglocentrismo del corpus e incluir *datasets* paralelos que trasciendan la lengua pívot.

Al ser un corpus centrado en lenguas con pocos o medios recursos, lo hemos seleccionado para cubrir dos de esos idiomas: el gallego y el suajili. Así, hemos realizado sendas particiones de entrenamiento, validación y test, manteniendo los tamaños del corpus Europarl[25].

4.2 Métricas de evaluación

La evaluación de los sistemas obtenidos para la traducción interactiva no es una cuestión trivial. No se está intentando estimar si la traducción que ofrece el sistema es mejor o peor. El propio traductor humano garantiza que la traducción obtenida sea de alta calidad. En realidad, un sistema de traducción interactiva será mejor cuanto menor sea el esfuerzo requerido por el usuario para obtener dicha traducción. Así se han seleccionado dos métricas que estiman el esfuerzo de traducción: la tasa de acciones de teclado y de ratón. Además, se ha estimado el BLEU y el TER de las primeras hipótesis ofrecidas para evaluar la calidad del modelo entrenado. A continuación se ofrece una descripción de las citadas métricas:

- El BLEU[32] es una métrica ampliamente utilizada para medir la calidad de las traducciones obtenidas automáticamente. Se trata de un conteo normalizado de la cantidad de *n-gramas* coincidentes entre la hipótesis de traducción y las referencias. Normalmente se utilizan valores para la n desde 1 a 4 y los resultados son combinados con una media geométrica y un factor de penalización para las oraciones más cortas, tal y como se indica en Papineni et al. [32]. En nuestro caso, evaluaremos la calidad de las traducciones que obtendría nuestro sistema sin intervención humana. Es decir, la calidad de las traducciones que se ofrecen en primera instancia al supervisor humano.

- La tasa de error de traducción o TER (por sus siglas e inglés) está muy relacionada con el error de edición. Más específicamente se define como:

  > "el mínimo número de ediciones necesarias para cambiar la hipótesis para que se corresponda exactamente con la referencia, normalizado por la longitud de la referencia"[38]

  Dichas ediciones toman la forma de inserción, borrado, sustitución o permutación de palabras.

- La tasa de correcciones de palabra o WSR (por sus siglas en inglés) mide el número de palabras que ha de teclear un supervisor humano, normalizado por el número de palabras de la traducción. En el ejemplo de la posedicion es igual al número de palabras incorrectas de la traducción ofrecida.

- La tasa de acciones de ratón o MAR (por sus siglas en inglés) mide el número de veces que el supervisor humano ha de usar el ratón para introducir algún cambio en la traducción. Esta métrica se suele normalizar por la cantidad de caracteres de la traducción final.

[5] https://opus.nlpl.eu/HPLT/corpus/version/HPLT

La obtención de la calidad BLEU es trivial, dado que solo hemos de obtener las traducciones directamente proporcionadas por cada LLM del conjunto de test. Para garantizar la consistencia con otros estudios hemos utilizado la implementación *sacrebleu* [34].

Para estimar la significatividad de las diferencias entre las métricas de los distintos sistemas hemos utilizado el test de aproximación aleatorizado propuesto por Riezler y Maxwell III [37]. De esta forma, podremos establecer si verdaderamente hay diferencias notorias entre los resultados aportados por los sistemas estudiados.

4.3 Simulación de la interacción con traductores humanos

Una manera de estimar el esfuerzo de traducción que implica un sistema es, directamente, realizar un experimento con traductores humanos que prueben el sistema desarrollado. Sin embargo, no es una solución que esté al alcance de todos, y tampoco de los autores de este trabajo. Es por ello que nosotros hemos realizado una simulación para aproximar dicho esfuerzo.

Así, para estimar el esfuerzo que dedicaría un traductor humano al usar nuestro sistema hemos necesitado realizar simulaciones de dicha interacción. Es decir, hemos inferido, a partir de una hipótesis y la frase de referencia, las acciones que realizaría un usuario real. Para ello, se han cruzado ambas cadenas de texto para obtener los segmentos comunes más largos que fueran exclusivos. Es decir, se ha forzado al sistema a obtener la traducción de referencia, validando los segmentos que comparte con cada una de las hipótesis. Por sencillez, la corrección ha sido siempre la palabra que debería sustituir a la primera palabra errónea de la hipótesis.

Para la simulación con el sistema basado en prefijos hemos realizado un procedimiento similar. Sin embargo, a la hora de realizar la realimentación solo hemos tenido en cuenta el segmento que cubre desde el inicio de la traducción hasta la primera palabra que no se corresponde con la referencia. Dicha primera palabra errónea ha sido la indicada como corrección.

Así, simplemente hemos debido contar las acciones que serían necesarias para realizar cada acción:

- La validación de un segmento nuevo de una sola palabra conlleva una acción de ratón. Si el segmento es más largo o es la extensión de otro previamente validado, se necesitan dos acciones de ratón para validarlo: una para indicar el inicio del segmento y otra para el final.
- Para indicar que dos segmentos deben permanecer unidos se necesitan dos acciones de ratón: una por cada segmento.
- Para indicar que un segmento debe colocarse al inicio de una traducción también se necesitan dos acciones de ratón: una para marcar el segmento y otra para indicar que la traducción ha de empezar por él.
- Para introducir una corrección se debe utilizar una acción de ratón, para colocarse en el lugar de dicha corrección, y otra acción de teclado, para introducirla.
- Para indicar al sistema que se debe realizar otra iteración se necesita una acción de ratón.
- Para indicar al sistema que la traducción ofrecida es correcta se necesita una acción de ratón.

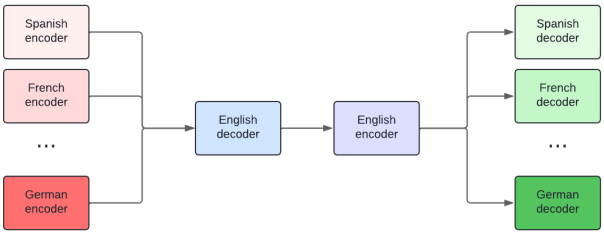


Figura 4.1: Arquitectura *Transformer Many-to-Many* con una lengua pívot. Es el caso del modelo *mBART* [39]

- Para indicar al sistema que la traducción adecuada es la resultante de unir todos los segmentos validados se necesita una acción de ratón.

Así, llevando la cuenta de todas las acciones realizadas pueden obtenerse el WSR y el MAR de la simulación. Puede observarse su implementación en el apartado de evaluación del repositorio generado para este proyecto[6].

4.4 Grandes modelos de lenguaje utilizados

Antes de explicar cómo se ha implementado el sistema y relatar qué resultados experimentales hemos obtenido, parece conveniente presentar los modelos que hemos utilizado. Estos modelos son el núcleo del sistema, sin los cuales el traductor humano tendría que escribir la traducción completamente a mano. Además, las diferencias que los caracterizan influirán en las prestaciones del sistema de traducción automática interactiva que los integre. Por tanto, parece ineludible el clarificar qué hace diferentes a estos modelos que constituyen uno de los pilares del trabajo realizado.

4.4.1. mBART

Los modelos mBART [39] son una familia de grandes modelos de lenguaje desarrollados por Facebook para la tarea de traducción automática. Utilizan una arquitectura codificador-decodificador basada en la tecnología *Transformer* [43]. Existen varias versiones, pero la que usaremos nosotros es la propuesta en Tang et al. [39], en el año 2020. Este modelo permite realizar traducciones entre múltiples idiomas y direcciones de traducción. Concretamente se cubren 50 lenguas que se extienden desde el Inglés al Nepalí, pasando por el Gallego. En Tang et al. [39] se describe cómo se usa una lengua intermedia o *pívot* para realizar la traducción entre idiomas diferentes del inglés. Este método se basa en traducir la frase de la lengua origen a la pívot, y después, de la lengua pívot a la destino tal y como puede apreciarse en la figura 4.1. De esta forma, implementando únicamente las direcciones de traducción que incluyen el idioma pívot, tenemos acceso a todas las direcciones de traducción entre los idiomas habilitados.

Por último, realizaremos algunos apuntes respecto de su facilidad de uso. El modelo de mBART que hemos usado consta de casi 611 millones de parámetros. En nuestros experimentos se ha necesitado de unos 4GB de memoria de la GPU para las pruebas de inferencia. El modelo está disponible para uso comercial y no comercial bajo la licencia Apache 2.0 y, por lo tanto, puede usarse tanto en ámbitos académicos como en aplicaciones con ánimo de lucro.

[6] https://github.com/sergiogg-ops/TFM_IMT

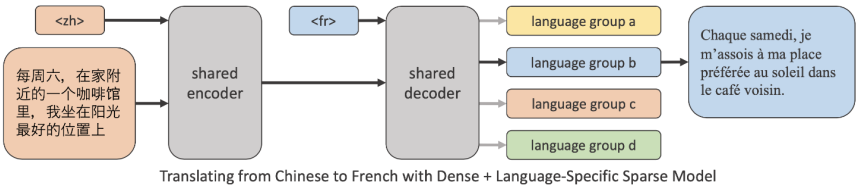


Figura 4.2: Imagen extraída de Fan et al. [12], donde se muestra el proceso de inferencia en un modelo *Transformer* con codificador y decodificador comunes para varias lenguas. Este es el caso del modelo *M2M100* [12].

4.4.2. M2M

Este modelo, propuesto por Fan et al. [12], supone un intento de reducir el tamaño de los modelos de traducción construidos por *Facebook* hasta ese momento. También está basado en la arquitectura codificador-decodificador de Vaswani et al. [43]. Sin embargo, al contrario que *mBART*, este modelo usa un mismo codificador y decodificador para todas las lenguas recogidas. Esta arquitectura es la conocida como *Many-to-Many* y puede observase su esquema en la figura 4.2, de ahí el nombre del modelo. En consecuencia, se consiguen aumentar considerablemente las direcciones de traducción permitidas. Para señalar el idioma origen al codificador se utiliza un *token* especial, junto al de inicio y final de secuencia. Para decodificar a la lengua destino se añade también este *token* de idioma al producto del codificador común. De esta forma, entre el codificador y el decodificador se crea una representación del significado de la oración origen válida para decodificar (traducir) a cualquier otro idioma habilitado.

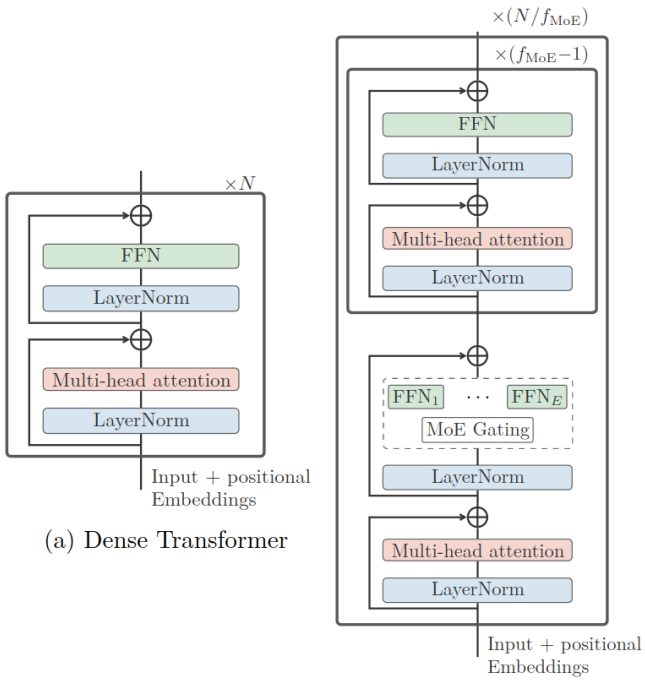
Con esta técnica, *M2M* consigue realizar traducciones entre 100 idiomas diferentes con 9900 direcciones de traducción distintas. Para ese fin, el modelo utilizado completo dispone de 418 millones de parámetros, aunque existe una versión mayor, de 1200 millones. Ambas versiones están disponibles bajo la licencia *MIT*, que permite su uso sin restricciones para "uso, copia, modificación, mezcla, publicación, distribución, sublicencia y/o venta de copias de Software"'. Sin embargo, el aviso de *copyright* y de permisos deben incluirse en las copias de suficiente tamaño.

4.4.3. Flan-T5

Esta familia de modelos está directamente emparentada con los modelos *T5* [36]. De hecho, para desarrollar los modelos *Flan-T5* [6] se usa directamente la arquitectura de dichos modelos y se aplica un *fine tuning* de tipo *FLAN* [45].

La nomenclatura *T5*[36] proviene de *Text-To-Text Transfer Transformer*, y toma la forma de una arquitectura *Transformer* muy similar a la descrita por Vaswani et al. [43]. Si bien es cierto que se implementan algunas diferencias. Por ejemplo, las capas de normalización no utilizan adición de sesgo. Esta familia de modelos fue entrenada con el corpus *C4* (*Colossal Clean Crawled Corpus*)[36], recogido expresamente para esta tarea.

El nombre *Flan* es un acrónimo de *Finetuned Language Net*[45], y constituye un proceso de ajuste de un modelo de lenguaje su uso por instrucciones. De esta forma, el objetivo es reentrenar un *LLM* para que las "respuestas" que genere a partir de los *prompts* obedezcan a las instrucciones contenidas en ellos. Este resultado se obtiene redefiniendo las muestras de diversos corpus de datos, precediéndolas de unos *prompts* que sean descriptivos de la tarea que pretende solucionar el corpus. Para más detalles puede consultarse el artículo de Wei et al. [45].



(a) Dense Transformer

(b) MoE Transformer

Figura 4.3: Imagen extraída de Costa-jussà et al. [8] donde se muestra la comparación entre la arquitectura *Transformer* original de Vaswani et al. [43] y la utilizada en el modelo *NLLB-200* de Costa-jussà et al. [8].

Así, la familia de modelos *Flan-T5* [6] supone la combinación de una arquitectura que ya había funcionado a *Google* con un proceso de *fine tuning* que hacía más versátil el modelo subyacente y ha sido entrenado sobre 60 lenguas distintas. Respecto de la interpretación para traducción automática podríamos describirlo como un modelo *many-to-many* con codificador y decodificador comunes para todas las lenguas. En este sentido, la dirección de traducción ha de codificarse como parte del *prompt* que precede a la oración origen. Actualmente, esta familia contiene cinco modelos de diferentes tamaños desde el pequeño (*small*) hasta el *xxl*. En el presente trabajo usaremos el base, con unos 250 millones de parámetros. Al igual que el modelo *mBART*, se distribuye bajo la licencia *Apache 2.0* y puede usarse con fines comerciales y no comerciales.

4.4.4. NLLB

Este modelo es fruto del proyecto *No Language Left Behind*[8] cuyo objetivo fue crear un único modelo de traducción para 210 idiomas. Desde un principio se busca proveer de un buen nivel de traducción para lenguas con pocos recursos. En este aspecto, una de las partes más importantes fue la creación del dataset de entrenamiento, denominado *FLORES-200*. La arquitectura del modelo *NLLB-200* se inspira en la propuesta por Vaswani et al. [43], pero incluye algunos módulos de *sparsely gated mixture of experts*, como puede verse en la figura 4.3. Cada vez que estas redes deben procesar una secuencia, se eligen los *k* expertos más apropiados para procesar cada token. Esta "decisión" la toma una red *feed foward* con activación *Softmax* que actúa de puerta. De esta forma se consigue aumentar la potencia del modelo sin que sea necesario utilizar todos sus parámetros en cada inferencia, lo que supone un ahorro de computación.

La versión del *NLLB-200* utilizada en este trabajo es un modelo más pequeño obtenido mediante *knowledge ditillation*[15], compuesto por 600 millones de parámetros. No

obstante, existen versiones más grandes de 1300 y 3300 millones de parámetros. Aún así, gracias a la naturaleza de la *SGMoE* no se usarán todos los parámetros al completo en la inferencia. El modelo se creó para favorecer la investigación en el area de traducción de idiomas con pocos recursos y está disponible bajo la licencia *Creative Commons Atribution Non Commercial 4.0 International*. Ello implica que puede usarse, reproducirse y compartirse, entre otros, con fines no comerciales.

4.5 Ajuste a la tarea de los modelos

Normalmente, antes de que un modelo neuronal sea funcional debe ajustarse para la tarea que nos gustaría que desarrollase. El proceso mediante el cual una red neuronal "aprende" a realizar dicha tarea es conocido como entrenamiento de la red. El método más habitual pasa por dividir el corpus de datos en tres particiones (o conjuntos) y usar para el entrenamiento dos de ellas: el conjunto de validación y el conjunto de aprendizaje. El usado para resolver el problema de optimización que plantea el ajustar el modelo neuronal a la tarea específica es el conjunto de aprendizaje. Este problema de optimización suele resolverse de forma iterativa usando algoritmos de descenso por gradiente. Así, para decidir cuándo no es necesario seguir el entrenamiento y prevenir el sobreajuste, se utiliza la partición de validación. En este segundo conjunto se prueba la red y se miden los resultados que obtiene. Si son lo suficientemente buenos, o simplemente no mejoran, el entrenamiento debería, a priori, detenerse. La tercera partición es denominada de test y sirve para estimar la calidad de la red ajustada en una fase posterior al entrenamiento. Existen protocolos de entrenamiento más sofisticados en los que aquí no nos extenderemos por exceder los objetivos del presente trabajo.

Los modelos que hemos utilizado para nuestros fines ya han sido preentrenados en ciertas tareas y se distribuyen con el ajuste propicio para resolverlas. De hecho, parte de su atractivo, como ya hemos descrito en otras secciones, es que ya sirven como modelos de lenguaje de los idiomas para los que han sido entrenados. Así, cuando son entrenados para la tarea de traducción que buscamos resolver, suelen obtener mejores resultados que los modelos entrenados desde la base en la tarea específica.

Siguiendo el procedimiento básico explicado, hemos realizado sendos reentrenamientos o *fine tuning* de los modelos para las distintas direcciones de traducción. De esta forma, hemos dividido los corpus aleatoriamente con los tamaños mostrados en el cuadro 4.1. Cuando se está realizando un *fine-tunings* es conveniente utilizar un factor de aprendizaje o *learning rate* más reducido, a fin de prevenir el olvido catastrófico [22]. Así, hemos utilizado diferentes *learning rates* de entre 10^{-5} y 10^{-7} en cada caso, utilizando los que mejores resultados obtenían. Además, se ha usado un *scheduler* lineal para reducir el *learning rate* a medida que avanza el entrenamiento. También hemos utilizado regularización de tipo *weight decay* de factor 10^{-2} para prevenir el sobreajuste. Como condición de parada hemos fijado la realización de tres pasadas completas al conjunto de aprendizaje o *epochs*. Adicionalmente, también hemos utilizado la técnica de *early stopping*[46] o parada temprana para evitar el sobreajuste y la prolongación excesiva del entrenamiento. Es decir, si el modelo no obtenía mejoras tras un número determinado de evaluaciones se ha detenido el entrenamiento. Finalmente, se han seleccionado los ajustes que mejor BLEU [32] obtuvieron sobre los conjuntos de evaluación.

Existen varios *frameworks* que permiten automatizar el entrenamiento y *fine tuning* de modelos neuronales. En este trabajo, comenzamos usando *Hugging Face*[7] puesto que es la principal plataforma de provisión de modelos preentrenados actualmente. Más tarde,

[7] https://huggingface.co/docs/transformers/training

combinamos la descarga de los modelos de *Hugging Face* con el *pipeline* de entrenamiento automatizado de *Pytorch Lightning*[8] por su versatilidad y facilidad de uso. Por último, destacar que realizamos todos los procesos aludidos sobre sendas máquinas remotas equipadas con *GPU Nvidia GeForce RTX-4090* y *GPU Nvidia Quadro RTX 8000*, cortesía del centro de investigación PRHLT[9].

4.6 Implementación del sistema

Un esquema de traducción puede plantearse de muchas formas. En principio podría utilizarse un mismo modelo para distintas direcciones de traducción. Esta aproximación sería, en principio, más eficiente respecto de los recursos necesarios. Sin embargo, utilizar modelos para direcciones de traducción específicas suele aportar mejores resultados. Es por ello por lo que nos hemos decantado en usar esta segunda opción. Así, primero, hemos realizado un fine-tuning para traducir oraciones de un idioma concreto a otro. Es decir, hemos ajustado un mismo LLM para cada dirección de traducción, creando modelos de traducción específicos. De esta forma, nuestro sistema usaría diferentes modelos en función de el idioma de la oración de entrada y el idioma deseado de la traducción de salida. Así, en cada iteración de traducción interactiva se dispone de una tupla de la forma:

$$S = (x, F, s, t, M) \tag{4.1}$$

Donde M se concreta como el modelo de traducción específico para la dirección de traducción deseada. Además, los idiomas origen y destino quedan representados por s y t respectivamente. La variable x se corresponde con la oración a traducir. Por último, F denota el conjunto de segmentos validados en alguna iteración anterior eliminando solapamientos. Así, el sistema mantiene en cada iteración de la traducción una memoria de las partes que ya han sido validadas y su orden relativo en la traducción. Es importante destacar que una vez ha sido validada una palabra se mantiene en el conjunto F hasta el final de la traducción. No obstante, el segmento que la contiene puede variar por la información que vaya aportando el usuario. Un mismo segmento puede extenderse, fusionarse con otro o, incluso, colocarse al inicio de la traducción. Es decir, el conjunto F es el único elemento de la tupla que se modifica entre iteraciones del proceso de traducción de una misma oración. De hecho, los cambios en este conjunto son los disparadores de la mejora progresiva de la traducción ofrecida.

El otro pilar del sistema es el algoritmo de búsqueda usado para explorar el espacio de posibles traducciones. En la aproximación de traducción neuronal suele utilizarse la búsqueda en haz como una forma subóptima de encontrar una buena solución. Nosotros también hemos usado este algoritmo de exploración con ciertos ajustes para respetar los confines del espacio reducido acotado en la ecuación (3.5). En este sentido, se ha atacado un problema de búsqueda en haz restringida. Por ello, se ha fijado una distancia máxima (en *tokens*) entre segmentos validados en la traducción obtenida.

Así, se ha realizado un control en cada paso de la generación con búsqueda en haz. En cada uno de dichos pasos se comprueba que la distancia desde el último *token* del último segmento introducido en la traducción no supere la máxima permitida. Si lo hace, se fuerza a generar el primer *token* del siguiente segmento. Además, si el último *token* formaba parte del segmento que se está introduciendo, se fuerza a introducir el siguiente *token* del segmento. En cualquier otro caso se deja libertad para escoger el token que indique la búsqueda en haz por defecto.

[8] https://lightning.ai/docs/pytorch/stable/
[9] https://www.prhlt.upv.es/

Además, se debe lidiar con los fragmentos de traducción que no están incluidos en el vocabulario de los modelos. Para tratar con estos casos, los modelos utilizan un *token* especial: el de palabra desconocida o <*unk*>. Si alguna parte de alguna restricción no pertenece al vocabulario del modelo, este no será capaz de generar dicha restricción por sí mismo. En estos casos, se ha tenido que hacer un posproceso de las hipótesis generadas para transcribir adecuadamente los *tokens* de palabra desconocida. En el caso que nos ocupa estas situaciones ocurren cuando el usuario introduce una palabra o carácter que no existe en el vocabulario del modelo. Aún en estas situaciones el sistema debe ser capaz de introducir dicho fragmento en su hipótesis de traducción. Por tanto, durante la búsqueda en haz se utiliza el *token* de palabra desconocida y, después, en la decodificación de *tokens* a lenguaje natural, se sustituye dicho token por su cadena de texto correspondiente.

El código *Python* utilizado para este proyecto está disponible en el repositorio correspondiente[10].

[10] https://github.com/sergiogg-ops/TFM_IMT

CAPÍTULO 5
Resultados experimentales

Tras presentar el fundamento teórico y las herramientas usadas procederemos, finalmente, a relatar los resultados del estudio empírico realizado. En primer lugar, nos centraremos en las lenguas con amplios recursos (español, francés y alemán con inglés) para describir la calidad de las traducciones de los modelos estudiados para dichos idiomas. Después presentaremos los resultados de dichos modelos integrados en sistemas basados en prefijos y segmentos. Posteriormente mostraremos los resultados de los mismos experimentos para las lenguas con pocos recursos (gallego y suajili con inglés). Por último, compararemos los resultados obtenidos en este trabajo con los obtenidos en estudios anteriores.

5.1 Lenguas con amplios recursos

Hemos realizado los mismos experimentos con tres pares de lenguas con muchos recursos, como lo son español-inglés, francés-inglés y alemán-inglés, y dos pares de lenguas con pocos recursos. Nuestro objetivo ha sido establecer si el esfuerzo estimado de traducción se ve influido por la cantidad de datos de la dirección de traducción que ha "visto" el modelo durante el entrenamiento. Así, primero nos centraremos en los tres pares de lenguas con muchos recursos, y analizaremos tanto la calidad de los modelos entrenados, como el esfuerzo de traducción estimado en sistemas de IMT basada en prefijos y segmentos.

5.1.1. Estudio de calidad de los modelos utilizados

Antes de evaluar el esfuerzo que requieren los distintos sistemas nos hemos propuesto estimar la calidad de las hipótesis de traducción ofrecidas por los modelos. Así, hemos realizado una pasada a los conjuntos de test para las distintas direcciones de traducción con todos los modelos entrenados. Los resultados, visibles en el cuadro 5.1, son un buen augurio del esfuerzo que requerirán después los sistemas que utilicen dichos modelos. Por lo pronto, podemos destacar que los mejores resultados de *BLEU* son los ofrecidos por el modelo *mBART*. No obstante, el modelo que ofrece una menor tasa de errores de traducción es el *M2M*. Aún así, hay pares de lenguas para los cuales las diferencias en ambas métricas no son tan significativas como para establecer si uno es superior al otro.

Parece bastante claro, que el modelo *NLLB* propone un mayor equilibrio entre *BLEU* y *TER*. Así, sus traducciones en ningún caso ofrecen las mejores métricas pero en todos los casos parece tener un *TER* más bajo que *mBART*. También ofrece un *BLEU* mejor cuando se le compara con *M2M*. De cara a la calidad de las traducciones ofrecidas, el peor modelo para los tres pares de lenguas es claramente *Flan-T5*. En ambas métricas queda a una gran

Tabla 5.1: Evaluación de las hipótesis iniciales del sistema sobre la partición de test del corpus Europarl[25]. Se han destacado en negrita los menores valores de esfuerzo para cada dirección de traducción. Además, se han señalado con un símbolo † aquellos pares de valores (dentro de la misma dirección de traducción) cuya diferencia no es estadísticamente significativa respectivamente.

Modelo	Origen	Destino	BLEU [↑]	TER [↓]
Flan-T5	de	en	21,5	61,4
	en	de	13,3	77,8
	en	es	17,7	61,8
	en	fr	30,8	56,4
	es	en	23,2	59,9
	fr	en	24,3	58,0
M2M	de	en	29,3	**51,1**
	en	de	26,4	**56,4**
	en	es	33,1	**47,7**
	en	fr	**38,8**†	**48,5**†
	es	en	30,5	**50,8**
	fr	en	33,7	**46,8**†
NLLB	de	en	31,0	53,0
	en	de	27,3	57,9
	en	es	34,3	49,4
	en	fr	39,1	50,9
	es	en	32,4	51,6
	fr	en	31,6	56,9
mBART	de	en	**37,7**	59,6
	en	de	**35,2**	64,7
	en	es	**40,3**	58,6
	en	fr	**40,5**†	**58,3**†
	es	en	**37,5**	60,2
	fr	en	**38,4**	**57,2**†

distancia de los otros tres modelos. Más adelante, mostraremos cómo repercute el buen ajuste del modelo a la dirección de traducción sobre el esfuerzo que necesitaría el usuario para utilizar el sistema.

5.1.2. Comparación de los sistemas basados en prefijos y en segmentos

Antes de realizar el estudio sobre el sistema de traducción interactiva basada en segmentos, parece conveniente compararlo con su competidor directo. Es decir, es especialmente relevante responder a la pregunta de si la flexibilidad aportada por el sistema basado en segmentos repercute en un menor esfuerzo humano frente al sistema basado en prefijos. En una comparación *a priori* entre ambos se deduce que el sistema basado en segmentos suponga un mayor uso del ratón. El sistema basado en prefijos solo utiliza este elemento para seleccionar un solo segmento y corregir la siguiente palabra. Sin embargo, con el sistema basado en segmentos, el usuario habrá de validar un número potencialmente mayor de segmentos. Además, existen otro tipo de acciones que también explotan el uso de este elemento, e.g. union de segmentos, colocación al inicio de un segmento, etc. Todo ello implica que, en nuestra suposición previa a la experimentación, el sistema basado en segmentos necesite un mayor uso del ratón.

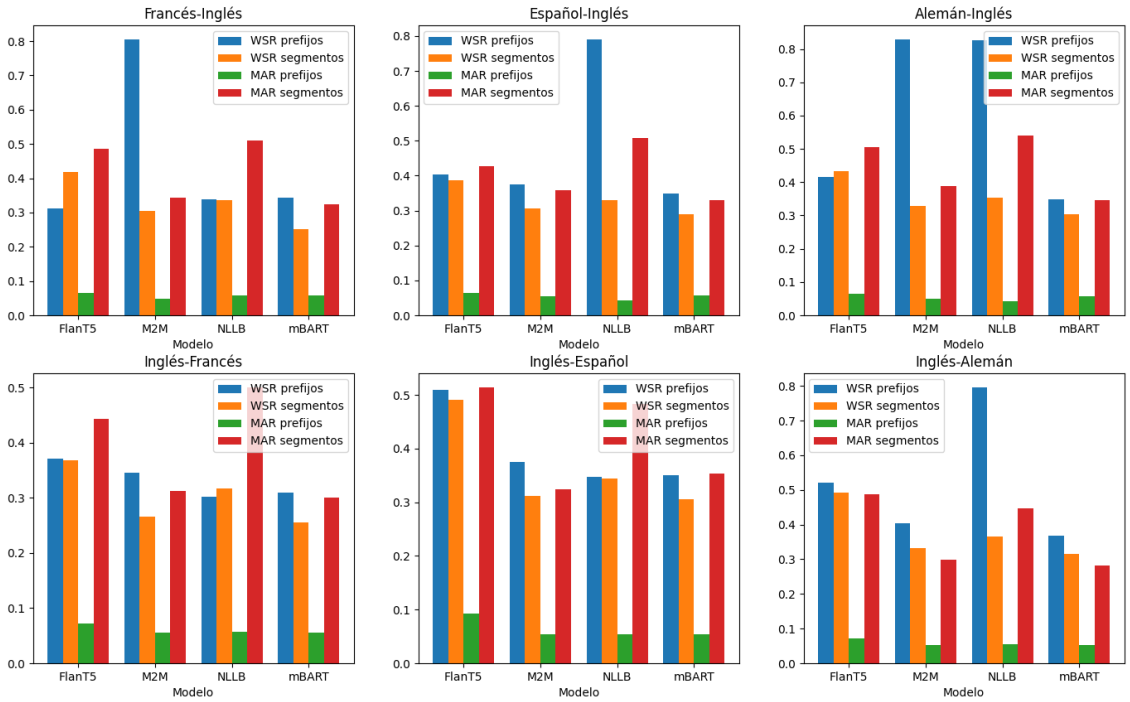


Figura 5.1: Comparación del esfuerzo de traducción empleado con el sistema de traducción basado en prefijos [29] y el basado en segmentos en los pares de lenguas con muchos recursos.

No obstante, el sistema basado en segmentos se propone para fijar partes de la traducción que son correctas y podrían cambiar con la aproximación basada en prefijos. Es decir, se pretende evitar el tener que introducir correcciones validando ciertas partes de la hipótesis de traducción. El resultado, debiera ser una reducción en el uso del teclado. Así, con nuestro análisis experimental pretendemos averiguar la magnitud del intercambio de esfuerzos que, *a priori*, se produciría.

Con ese objetivo, se utilizaron los mismos modelos con un sistema basado en prefijos y otro en segmentos. El esfuerzo de traducción estimado con estos sistemas para cada dirección de traducción están disponibles en la figura 5.1. Una vez hemos realizado la experimentación podemos corroborar que, efectivamente, se produce el desvío de esfuerzo del teclado al ratón. Si bien en algunos casos anómalos aumenta el uso de ambos instrumentos, e.g. el modelo *Flan-T5* para la traducción de alemán a inglés.

No obstante, al menos para estas lenguas, la disminución en el uso del teclado no parece tan drástica como el aumento del uso del ratón. Estos resultados contrastan con los obtenidos en otros estudios previos como el de Domingo, Peris y Casacuberta [10]. En dicho estudio se utilizaron métodos de traducción automática estadística, con los que la aproximación basada en segmentos el desplazamiento de esfuerzo se hacía de forma más o menos limpia, tal y como hemos concebido *a priori*. También como explicaremos en el apartado 5.4, nuestros sistemas basados en prefijos suponen un esfuerzo considerablemente menor para el usuario. Sin embargo, esa mejora no se ha trasladado también a los sistemas neuronales basados en segmentos para estos pares de lenguas.

5.1.3. Resultados de los diferentes modelos con el sistema basado en prefijos

Una vez comparados ambos sistemas vamos a proceder a analizar el esfuerzo estimado que requieren los diferentes modelos con el sistema de IMT basada en prefijos. Observando las métricas expuestas en el cuadro 5.2, no podemos declarar un modelo co-

Tabla 5.2: Esfuerzo de traducción estimado para los sistemas basados en prefijos [29] con los diferentes modelos y los pares de lenguas con muchos recursos. Se han destacado en negrita los menores valores de esfuerzo para cada dirección de traducción. Además, se han señalado con un símbolo † aquellos pares de valores (dentro de la misma dirección de traducción) cuya diferencia no es estadísticamente significativa respectivamente.

Modelo	Origen	Destino	WSR [↓]	MAR [↓]
Flan-T5	de	en	41,4	6,5
	en	de	52,0	7,1
	en	es	50,9	9,2
	en	fr	37,1	7,2
	es	en	40,2	6,5
	fr	en	**31,2**†	6,6
M2M	de	en	83,0	4,9
	en	de	40,4	5,3
	en	es	37,5	**5,4**†
	en	fr	34,6	**5,5**
	es	en	37,5	5,4
	fr	en	80,6	**4,7**
NLLB	de	en	82,6	**4,2**
	en	de	79,7	**5,5**†
	en	es	**34,8**	**5,4**†
	en	fr	**30,3**	5,7†
	es	en	79,1	**4,2**
	fr	en	33,9	5,7
mBART	de	en	**34,8**	5,7
	en	de	**36,8**	**5,2**†
	en	es	35,1	**5,4**†
	en	fr	31,0	5,5†
	es	en	**34,8**	5,7
	fr	en	**34,3**†	5,7

mo el mejor en todos los pares de lenguas. De hecho, hay numerosos pares de lenguas y modelos para los que no hemos hallado significatividad estadística suficiente como para diferenciar cuál funciona mejor. Ocurre sobretodo entre los modelos *mBART* y *NLLB*, que agrupan la mayor parte de los primeros puestos. Un análisis más profundo reporta que, en las lenguas en las que *NLLB* no funciona extraordinariamente bien, implica una tasa de uso del teclado exorbitada. Es el caso de la traducción entre alemán e inglés y de español a inglés. No obstante, estos son los casos en los que necesita un menor uso del ratón. De *mBART* podemos decir que es el modelo que, en promedio, funciona mejor para todas las lenguas sin que haya ninguna en los que obtenga resultados catastróficos.

Con el modelo *M2M* ocurre algo similar que con el *NLLB*. En algunas lenguas ofrece una tasa de esfuerzo general competitiva, mientras que en la traducción de alemán a inglés y de francés a inglés se dispara el uso del teclado. También en estos dos casos obtiene su menor tasa de uso del ratón.

Finalmente, el modelo *Flan-T5* necesita, en general, mayor esfuerzo que el resto de modelos utilizados. El uso de ratón que implica es siempre disminuido por algún otro modelo. Sin embargo, sorprendentemente es el modelo que menos uso del teclado necesita para la traducción de francés a inglés, junto con *mBART*.

Tabla 5.3: Esfuerzo de traducción estimado para los sistemas basados en segmentos con los diferentes modelos y los pares de lenguas con muchos recursos. Se han destacado en negrita los menores valores de esfuerzo para cada dirección de traducción. Además, se han señalado con un símbolo † o ‡ aquellos pares de valores (dentro de la misma dirección de traducción) cuya diferencia no es estadísticamente significativa respectivamente.

Modelo	Origen	Destino	WSR [↓]	MAR [↓]
Flan-T5	de	en	43,3	50,6
	en	de	49,2	48,8
	en	es	49,1	51,5
	en	fr	36,8	44,3
	es	en	38,8	42,7†‡
	fr	en	41,7	48,6
M2M	de	en	33,0	38,8
	en	de	33,3†	29,9
	en	es	31,2†	**32,4**
	en	fr	26,6	31,3
	es	en	30,7	35,8‡
	fr	en	30,5†	34,3
NLLB	de	en	35,4	54,0
	en	de	36,5†	44,6
	en	es	34,4†	48,3
	en	fr	31,6	50,1
	es	en	33,0	50,7
	fr	en	33,6†	51,0
mBART	de	en	**30,3**	**34,6**
	en	de	**31,6**	**28,1**
	en	es	**30,5**	35,3
	en	fr	25,6	30,1
	es	en	**29,0**	**33,0**†
	fr	en	**25,2**	**32,4**

5.1.4. Resultados de los diferentes modelos con el sistema basado en segmentos

Finalmente, vamos a proceder a comparar los sistemas de traducción automática interactiva basada en segmentos con los distintos modelos. A la luz de las métricas expuestas en el cuadro 5.3 podemos decir que el modelo que menos esfuerzo supone para el usuario es el *mBART* en la gran mayoría de las direcciones con muchos recursos probadas. No obstante, también es cierto que para la traducción de inglés a español el modelo *M2M* parece implicar una tasa menor de uso del ratón. Lo cierto es que este modelo es un buen competidor de *mBART*, al menos para estos tres pares de lenguas. Sus tasas de esfuerzo son mucho más cercanas al primer puesto que las de *Flan-T5* o *NLLB*. Respecto de este último, en la simulación ha supuesto una tasa de uso del teclado mayor que los otros dos modelos pero, aún así, comparable. No obstante, el uso estimado del ratón es mucho mayor para los tres pares de lenguas. Esta tasa de uso, es comparable e, incluso mayor en algunos casos, a la del modelo *Flan-T5*. Sin embargo, este último modelo necesita un uso del teclado desorbitado que supera con creces al del resto de modelos probados.

A pesar de que en las métricas medias de esfuerzo el modelo *Flan-T5* es el peor parado, un análisis más profundo puede revelar algo sorprendente. Al realizar un estudio de la significatividad de la tasa de uso del ratón, hemos comprobado que, para la traducción de español a inglés, no existe una diferencia suficientemente importante entre el modelo

Flan-T5 y los modelos *mBART* y *M2M*. En los demás casos, no hay dudas razonables de su inferioridad en esta tarea. El mismo estudio revela que en algunos casos los modelos *M2M* y *NLLB* implican una tasa de uso del teclado bastante similar.

Por último, destacaremos que cinco de las seis direcciones de traducción, para el mismo modelo, han obtenido resultados comparables respecto del esfuerzo necesario para su uso. Sin embargo, la traducción de inglés a francés parece necesitar de un menor esfuerzo para todos los modelos estudiados. La única excepción es la tasa de uso de ratón para el modelo *NLLB*, que es mayor que para otras direcciones.

5.2 Lenguas con pocos recursos

Tras haber analizado los resultados obtenidos para las lenguas con muchos recursos vamos a proceder a hacer lo propio con las de pocos recursos. Recordemos que nuestro objetivo es establecer si cuando el modelo está más pobremente entrenado en una dirección de traducción se produce algún cambio en el esfuerzo requerido por un usuario de un sistema de IMT.

5.2.1. Estudio de calidad de los modelos utilizados

Al igual que con los pares de lenguas con muchos recursos, hemos traducido los conjuntos de test del corpus con los diferentes modelos sin traducción interactiva. Nuevamente buscamos saber si los modelos mejor entrenados implican un menor esfuerzo una vez se usan en el sistema de traducción interactiva. Así, fuera de toda duda el modelo *M2M* es el que parece dar mejores traducciones para los pares de lenguas utilizados. Tanto los valores de *BLEU* como los de *TER* por ellos son significativamente menores que los del resto de modelos para casi todas las direcciones de traducción. La única excepción es para las traducciones del gallego al inglés, en la que la diferencia en el TER con *mBART* no es lo suficientemente significativa como para tenerla en cuenta. Precisamente el modelo *mBART* también ofrece unas métricas razonables en comparación con las de *M2M*. No obstante, para la traducción del suajili al inglés su error de traducción se dispara considerablemente.

Tanto el modelo *Flan-T5*[6] como, sorprendentemente, el *NLLB* ofrecen traducciones significativamente peores al par mencionado previamente. Los valores de *BLEU* de sus traducciones son inferiores a la mitad del mejor modelo y los de *TER* son simplemente desmesurados. Si bien *Flan-T5* parece funcionar ligeramente mejor para el par inglés-gallego y *NLLB* para el par inglés-suajili.

Así, tenemos dos modelos que, *a priori*, parecen preparados para la tarea (*mBART* y *M2M*) y otros dos que funcionan considerablemente mal (*Flan-T5* y *NLLB*).

5.2.2. Comparación de los sistemas basados en prefijos y en segmentos

En traducción interactiva con idiomas con bajos recursos los modelos utilizados parecen dividirse en dos subgrupos. En el primero de ellos, ocurre algo parecido a las lenguas con muchos recursos. Es decir, nuevamente los sistemas basados en prefijos ofrecen una tasa de uso de ratón que, generalmente, se reduce ligeramente con los sistemas basados en segmentos. Además, vemos otra vez como el drástico aumento del uso del ratón que implican estos últimos no compensa la disminución del esfuerzo de teclado. Tanto los modelos *Flant-T5* como los *NLLB* forman parte de este primer conjunto de modelos. Así,

Tabla 5.4: Evaluación de las hipótesis iniciales del sistema sobre la partición de test del corpus HPLT[9]. Se han destacado en negrita los menores valores de esfuerzo para cada dirección de traducción. Además, se han señalado con un símbolo † aquellos pares de valores (dentro de la misma dirección de traducción) cuya diferencia no es estadísticamente significativa respectivamente.

Modelo	Origen	Destino	BLEU [↑]	TER [↓]
Flan-T5	en	gl	10,0	79,1
	en	sw	10,9	194,7
	gl	en	20,7	60,4
	sw	en	7,3	759,4
M2M	en	gl	**33,2**	**37,2**
	en	sw	**52,0**	**39,3**
	gl	en	**34,4**	**38,1**†
	sw	en	**52,6**	**35,8**
NLLB	en	gl	4,2	322,2
	en	sw	10,8	92,9
	gl	en	13,4	269,5
	sw	en	23,4	94,7
mBART	en	gl	29,5	40,5
	en	sw	44,2	88,9
	gl	en	33,4	**38,8**†
	sw	en	50,5	53,6

en esta división, podemos establecer que los modelos *NLLB* integrados en sistemas de traducción interactiva basada en prefijos son los que menor esfuerzo suponen al usuario.

Como integrantes del segundo grupo de modelos tenemos a los *mBART* y *M2M* con un comportamiento un tanto diferente. Estos modelos muestran una tasa de uso de teclado anómalamente alta con los sistemas de traducción interactiva basada en prefijos. Si bien, es cierto que el uso del ratón se mantiene igualmente bajo. También con estos modelos el uso de dicho instrumento aumenta considerablemente cuando se integran en sistemas basados en segmentos. Sin embargo, en este caso, se produce una disminución en el uso del teclado en igual medida. Es decir, para estos pares de lenguas, con los modelos *mBART* y *M2M* usar la variante de traducción interactiva basada en segmentos puede ser un acierto en función de las preferencias del usuario.

Con todo ello, parece que en las lenguas que estamos estudiando la mejor aproximación es el uso de un sistema de traducción interactiva basada en prefijos que integre un modelo *NLLB*. Es el que menos esfuerzo total implica por parte del usuario. Si bien es cierto que tiene mayor tasa de uso del teclado que otras combinaciones, su bajo uso del ratón hace que compense con creces esta diferencia.

5.2.3. Resultados de los diferentes modelos con el sistema basado en prefijos

Entre los sistemas de IMT basada en prefijos para nuestros pares de lenguas de bajos recursos destacan los que incluyen el modelo *NLLB*. Para todas las direcciones de traducción supone un esfuerzo de traducción menor para el usuario que con ningún otro modelo. Dicho menor grado de esfuerzo se materializa tanto en el uso del teclado como del ratón, tal y como se puede apreciar en el cuadro 5.5.

Parece interesante observar cómo los modelos *M2M*, a pesar de proporcionar unas mejores primeras hipótesis, implica un esfuerzo de traducción, en ocasiones, incluso mayor que los *Flan-T5*. Gran parte de este esfuerzo se debe al uso del teclado. De hecho, si nos centramos únicamente en el esfuerzo dedicado al ratón, los modelos *M2M* son com-

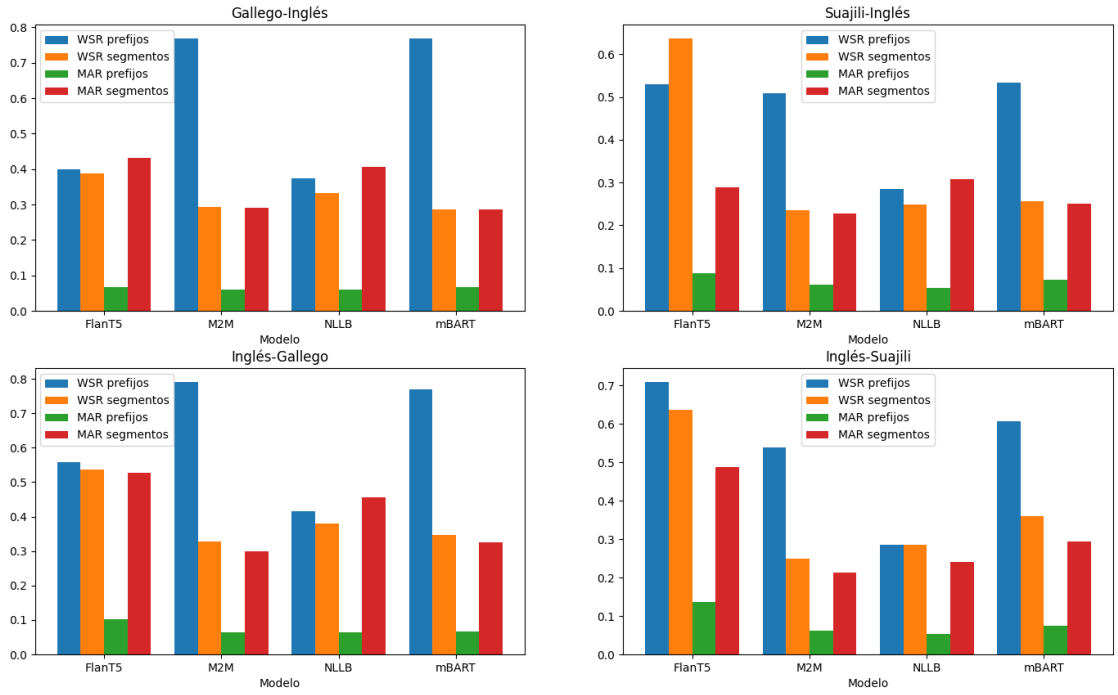


Figura 5.2: Comparación del esfuerzo de traducción empleado con el sistema de traducción basado en prefijos [29] y el basado en segmentos en los pares de lenguas con pocos recursos.

petitivos con los *NLLB* y mejoran los *Flan-T5*. No obstante, la tasa de uso de teclado es desorbitada para el par inglés-gallego, y bastante mala para el inglés-suajili. Todo ello favorece que, a pesar de ser un buen modelo para traducción automática simple, en la tarea de traducción automática interactiva basada en prefijos para los pares de lenguas tratados en esta sección sea bastante mediocre. Con los modelos *mBART* ocurre algo parecido. De hecho, los valores de sus tasas de esfuerzos son muy similares. Recordemos que también este último modelo obtenía unas primeras hipótesis de una calidad mucho mayor a los modelos *NLLB* y *Flan-T5*.

5.2.4. Resultados de los diferentes modelos con el sistema basado en segmentos

Respecto de la traducción automática interactiva basada en segmentos para las lenguas de pocos recursos el modelo *M2M* parece ejercer una dominación absoluta. En los dos pares de lenguas es el que menor esfuerzo del usuario necesita para obtener las traducciones adecuadas tal y como se aprecia en el cuadro 5.6. Únicamente para la traducción de gallego a inglés parece ser igualado por el *mBART*, puesto que la diferencia no es estadísticamente significativa en el uso del ratón ni del teclado.

A pesar de que *mBART* sobrepasa al resto de modelos en la dirección gallego-inglés, no sería razonable decir que es el segundo mejor modelo. El esfuerzo que requeriría un usuario con el modelo *NLLB* para traducir entre el inglés y el suajili (en cualquiera de las dos direcciones) sería igual o menor. Con todo ello, para el par de lenguas inglés-gallego el *mBART* sí parece requerir un menor esfuerzo que *NLLB* y *Flan-T5*.

Recordemos que, en lo que a la calidad de las primeras hipótesis se refiere, los modelos *NLLB* y *Flan-T5* eran absolutamente inadecuados para la tarea. Sin embargo, cuando se usan en un sistema de traducción interactiva basada en prefijos se comportan de forma distinta. Mientras el *Flan-T5* requiere considerablemente más esfuerzo, el modelo *NLLB*

Tabla 5.5: Esfuerzo de traducción estimado para los sistemas basados en prefijos[29] con los diferentes modelos y los pares de lenguas con pocos recursos. Se han destacado en negrita los menores valores de esfuerzo para cada dirección de traducción. Además, se han señalado con un símbolo † aquellos pares de valores (dentro de la misma dirección de traducción) cuya diferencia no es estadísticamente significativa respectivamente.

Modelo	Origen	Destino	WSR [↓]	MAR [↓]
Flan-T5	en	gl	55,8	10,2
	en	sw	71,0	13,6
	gl	en	**39,9**†	6,8
	sw	en	52,9†	8,8
M2M	en	gl	79,1	6,4
	en	sw	54,0	6,1
	gl	en	76,9‡	6,0
	sw	en	51,0	6,1
NLLB	en	gl	**41,6**	**6,3**
	en	sw	**28,5**	**5,3**
	gl	en	**37,5**†	**6,0**
	sw	en	**28,6**	**5,3**
mBART	en	gl	76,95	6,62
	en	sw	63,63	7,44
	gl	en	77,0‡	6,6
	sw	en	53,48†	7,32

es competitivo. Nuevamente, destacamos que, a pesar de ofrecer unas primeras hipótesis de traducción notoriamente pobres, iguala e incluso supera al modelo *mBART* en lo que a esfuerzo del usuario se refiere.

Precisamente con los modelos *NLLB* y *Flan-T5* podemos observar un fenómeno interesante: los pares de lenguas en los que el modelo ofrecía mejores primeras hipótesis son los que menos esfuerzo del usuario suponen. Así, el modelo *Flan-T5* necesita menos esfuerzo del usuario para las traducciones que involucran el gallego, mientras que con el *NLLB* ocurre lo mismo para las traducciones entre el suajili y el inglés.

5.3 Ejemplos reales de funcionamiento

Hasta ahora hemos visto a nivel general cómo funcionan nuestros modelos. Nos hemos valido de las métricas y estadísticas para evaluar todas las combinaciones expuestas. Sin embargo, nos gustaría dar ciertas nociones de qué significan estas estadísticas con ejemplos reales. Así, vamos a exponer unos cuantos ejemplos de simulaciones de interacción entre el traductor humano y el sistema con sus correspondientes métricas para esclarecer qué representan exactamente.

Así, en la figura 5.3 podemos apreciar un ejemplo de traducción en la que el paradigma basado en prefijos funciona considerablemente bien. En dicho ejemplo, la traducción se resuelve en tan solo dos iteraciones. Solo es necesario introducir una corrección y validar el prefijo señalado en la iteración 1. Podemos ver que con ese ligero cambio el modelo es capaz de producir la traducción sin problemas. Es decir, el modelo está lo suficientemente bien entrenado como para obtener la traducción correcta con pocas correcciones.

No obstante, esta aproximación no siempre funciona tan bien. En ocasiones, como en el ejemplo mostrado en la figura 5.4, necesita demasiadas correcciones para producir la traducción correcta. Incluso podemos ver fragmentos como "*veiled homophobia*" que,

Tabla 5.6: Esfuerzo de traducción estimado para los sistemas basados en segmentos con los diferentes modelos y los pares de lenguas con pocos recursos. Se han destacado en negrita los menores valores de esfuerzo para cada dirección de traducción. Además, se han señalado con un símbolo † aquellos pares de valores (dentro de la misma dirección de traducción) cuya diferencia no es estadísticamente significativa.

Modelo	Origen	Destino	WSR [↓]	MAR [↓]
Flan-T5	en	gl	53,7	58,2
	en	sw	63,8	72,5
	gl	en	38,7	48,0
	sw	en	63,8	72,5
M2M	en	gl	**32,8**	**38,4**
	en	sw	**25,0**	**26,4**
	gl	en	**29,3**†	**36,6**†
	sw	en	**23,5**	**27,7**
NLLB	en	gl	38,0	45,6
	en	sw	28,5	29,1
	gl	en	33,1	41,0
	sw	en	24,9	30,8*
mBART	en	gl	34,6	41,7
	en	sw	35,9	37,6
	gl	en	**28,7**†	**35,8**†
	sw	en	25,6	30,7†

Oración de origen: le mode de vie qu apos;il a défendu avec ténacité contre le fascisme
Traducción de referencia: the way of life that he defended so tenaciously against fascism

Iter. 0	MT	the way of life that he tenaciously defended against fascism
Iter. 1	Usuario	the way of life that he **defended** tenaciously defended against fascism
	MT	the way of life that he defended so tenaciously against fascism
Iter. 2	Usuario	the way of life that he defended so tenaciously against fascism
	MT	the way of life that he defended so tenaciously against fascism

Figura 5.3: Traza del proceso de traducción interactiva basada en prefijos sobre las frases de origen y referencia indicadas. Dicho proceso necesitó de un esfuerzo correspondiente a un WSR de 20,0 y un MAR de 6,0. Dentro de cajas se muestra el prefijo validado y en negrita las correcciones introducidas.

Oración de origen: D'abord vous utilisez des arguments spécieux, puis vous les discréditer par une homophobie à peine voilé.
Traducción de referencia: You raise a straw man and then knock it down with thinly veiled homophobia.

Iter. 0	MT	First you use specious arguments and then discredit them with barely veiled homophobia.
Iter. 1	Usuario	**You** First you use specious arguments and then discredit them with barely veiled homophobia.
	MT	You First, you use specious arguments, and then you discredit them with barely veiled homophobia.
Iter. 2	Usuario	You **raise** First, you use specious arguments, and then you discredit them with barely veiled homophobia.
	MT	You raise use specious arguments first, and then you discredit them with barely veiled homophobia.
Iter. 3	Usuario	You raise **a** use specious arguments first, and then you discredit them with barely veiled homophobia.
	MT	You raise a specious arguments and then discredit them with a barely veiled homophobia.
Iter. 4	Usuario	You raise a **straw** specious arguments and then discredit them with a barely veiled homophobia.
	MT	You raise a straw straitjacket and then discredit it with a barely veiled homophobia.
Iter. 5	Usuario	You raise a straw **man** straitjacket and then discredit it with a barely veiled homophobia.
	MT	You raise a straw man , and then you discredit it with a barely veiled homophobia.
Iter. 6	Usuario	You raise a straw man **and**, and then you discredit it with a barely veiled homophobia.
	MT	You raise a straw man and 's arguments and then discredit them with a barely veiled homophobia.
Iter. 7	Usuario	You raise a straw man and **then**'s arguments and then discredit them with a barely veiled homophobia.
	MT	You raise a straw man and then then discredit it with a barely veiled homophobia.
Iter. 8	Usuario	You raise a straw man and then **knock** then discredit it with a barely veiled homophobia.
	MT	You raise a straw man and then knock discredit it with a barely veiled homophobia.
Iter. 9	Usuario	You raise a straw man and then knock **it** discredit it with a barely veiled homophobia.
	MT	You raise a straw man and then knock it it down with a barely veiled homophobia.
Iter. 10	Usuario	You raise a straw man and then knock it **down** it down with a barely veiled homophobia.
	MT	You raise a straw man and then knock it down down with a barely disguised homophobia.
Iter. 11	Usuario	You raise a straw man and then knock it down **with** down with a barely disguised homophobia.
	MT	You raise a straw man and then knock it down with with a barely veiled homophobia.
Iter. 12	Usuario	You raise a straw man and then knock it down with **thinly** with a barely veiled homophobia.
	MT	You raise a straw man and then knock it down with thinly , naked homophobia.
Iter. 13	Usuario	You raise a straw man and then knock it down with thinly veiled, naked homophobia.
	MT	You raise a straw man and then knock it down with thinly veiled veiled homophobia.
Iter. 14	Usuario	You raise a straw man and then knock it down with thinly veiled homophobia veiled homophobia.
	MT	You raise a straw man and then knock it down with thinly veiled homophobia pie.
Iter. 15	Usuario	You raise a straw man and then knock it down with thinly veiled homophobia . pie.
	MT	You raise a straw man and then knock it down with thinly veiled homophobia. anyway.
Iter. 15	Usuario	You raise a straw man and then knock it down with thinly veiled homophobia. anyway.
	MT	You raise a straw man and then knock it down with thinly veiled homophobia.

Figura 5.4: Traza del proceso de traducción interactiva basada en prefijos sobre las frases de origen y referencia indicadas. Dicho proceso necesitó de un esfuerzo correspondiente a un WSR de 100 y un MAR de 9,2. Dentro de cajas se muestra el prefijo validado y en negrita las correcciones introducidas.

Oración de origen: Le tourisme de santé, tel que défini par les cinglés de droite, n'existe pratiquement pas.
Traducción de referencia: Health tourism as defined by the right wing loonies is virtually none existent.

Iter. 0	MT	Health tourism, as defined by the right wing wing, hardly exists.
Iter. 1	Usuario	Health tourism, as defined by the right wing **loonies** hardly exists .
	MT	Health tourism as defined by the right wing wing virtual loonies s .
Iter. 2	Usuario	Health tourism as defined by the right wing ~~wing virtual~~ loonies **is** s .
	MT	Health tourism as defined by the right wing loonies is ly absent .
Iter. 3	Usuario	Health tourism as defined by the right wing loonies is **virtually** ly absent .
	MT	Health tourism as defined by the right wing loonies is virtually -existent .
Iter. 4	Usuario	Health tourism as defined by the right wing loonies is virtually **none** - existent .
	MT	Health tourism as defined by the right wing loonies is virtually none existent.

Figura 5.5: Traza del proceso de traducción interactiva basada en segmentos sobre las frases de origen y referencia indicadas. Dicho proceso necesitó de un esfuerzo correspondiente a un WSR de 28,6 y un MAR de 25,0. Dentro de cajas se muestran los segmentos validados, en negrita las correcciones introducidas y los fragmentos tachados indican unión de los segmentos validados que los delimitan.

siendo correctos, no se mantienen siempre en la traducción. Este es el comportamiento que se pretendía evitar con la traducción interactiva basada en segmentos. Por lo demás, para este ejemplo ha de introducirse la traducción entera por teclado, lo que no supone ninguna ventaja a la hora de usar IMT.

Si nos fijamos en la figura 5.5 podemos observar una traducción que el sistema de traducción interactiva basado en segmentos resuelve efectivamente en cuatro iteraciones. Es cierto que el primer segmento señalado es convenientemente grande, pero también podemos apreciar cómo se validan segmentos poteriores para fijarlos como parte de la traducción deseada. Además, también se utiliza la unión de segmentos para evitar la intrusión de palabras que no devieran colocarse entre ellos. También vemos cómo, en la última iteración, el usuario indica al sistema que la traducción deseada es la que se obtendría de concatenar todos los segmentos validados. En resumen, es un ejemplo donde se utilizan muchas de las posibilidades que ofrece la traducción interactiva basada en prefijos de manera muy efectiva.

Sin embargo, en otras traducciones, como la de la figura 5.6, este sistema funciona considerablemente peor. Dicho ejemplo es especialmente complicado debido en gran parte a los apóstrofes que están compuestos por varios *tokens*. Sin embargo, podemos observar cómo, mientras se corrige el primero, el segundo se corrige casi por completo automáticamente. Otro factor que aumenta el esfuerzo es que a menudo se introducen nuevas palabras entre segmentos que deberían estar unidos. La aplicación de esta corrección también aumenta el esfuerzo, y más cuando también se debe extender alguno de esos segmentos. También se utiliza la posibilidad de fijar un segmento al inicio, cómo puede observarse en la iteración 2 de la figura 5.6. En realidad, en este ejemplo se utilizan casi todas las posibilidades que ofrece el paradigma basado en segmentos. Por ello, al realizar tantas acciones el esfuerzo necesario para traducir dicha oración es tan elevado.

5.4 Comparación con estudios anteriores

Tal y como se ha comentado en capítulos anteriores se han entrenado varios grandes modelos de lenguaje con el corpus *Europarl* [25]. Así mismo, se ha evaluado el sistema

Oración de origen: Ne vous inquiétez pas, je suis sûr que les commentaires vont bientôt s'enflammer.
Traducción de referencia: Don' t worry , I' m sure the comments section will take off soon.

Iter. 0	MT	I am sure that the comments will soon be blown up.
Iter. 1	Usuario	**Don** I am sure ~~that~~ the comments will soon ~~be blown up~~ .
	MT	I am Don I , I sure the comments will soon.
Iter. 2	Usuario	~~I am~~ Don **&** I , I sure the comments will soon.
	MT	Don & , I & sure the comments will soon.
Iter. 3	Usuario	Don & **apos** , I& sure the comments will soon.
	MT	Don & apos , I& apos sure the comments will soon.
Iter. 4	Usuario	Don & apos **;** , I' sure the comments will soon.
	MT	Don & apos ; worry , I' ' sure the comments will soon.
Iter. 5	Usuario	Don & apos ; **t** worry, I' ' sure the comments will soon.
	MT	Don & apos ; worry t ' worry, I' ' sure the comments will soon.
Iter. 6	Usuario	Don & apos ; ~~worry~~ t ~~'~~ worry, I' **;** ' sure the comments will soon.
	MT	Don & apos ;t worry, I' ; m sure the comments will soon.
Iter. 7	Usuario	Don & apos ;t worry, I' ;m sure the comments **section** will soon.
	MT	Don & apos ;t worry, I' ;m sure the comments soon be section . will ' soon.
Iter. 8	Usuario	Don & apos ;t worry, I' ;m sure the comments ~~soon be~~ section ~~.~~ will **take**' soon.
	MT	Don & apos ;t worry, I' ;m sure the comments section will be take a turn soon.
Iter. 9	Usuario	Don & apos ;t worry, I' ;m sure the comments section will be take **off** a turn soon.
	MT	Don' t worry , I' m sure the comments section will take off soon.

Figura 5.6: Traza del proceso de traducción interactiva basada en segmentos sobre las frases de origen y referencia indicadas. Dicho proceso necesitó de un esfuerzo correspondiente a un WSR de 42,9 y un MAR de 85,5. Dentro de cajas se muestran los segmentos validados, en negrita las correcciones introducidas y los fragmentos tachados indican unión de los segmentos validados que los delimitan.

Tabla 5.7: Extracto de los resultados de los experimentos coincidentes con el estudio de Domingo, Peris y Casacuberta [10]. En dicho estudio se emplearon modelos de traducción estadísticos sobre sistemas de IMT basadas en prefijos y en segmentos.

				Prefijos		Segmentos	
Origen	Destino	BLEU[↑]	TER[↓]	WSR[↓]	MAR[↓]	WSR[↓]	MAR[↓]
de	en	19,2	61,1	73,3	17,7	34,4	30,8
en	de	15,3	68,4	75,0	15,0	33,1	25,9
en	fr	26,5	55,6	61,4	13,5	31,5	28,4
fr	en	26,5	51,4	58,7	13,9	32,7	30,3

Tabla 5.8: Extracto de los resultados de los experimentos coincidentes con el estudio de Peris, Domingo y Casacuberta [33]. En dicho estudio se emplearon modelos neuronales sin preentrenamiento, basados en la arquitectura de Bahdanau, Cho y Bengio [3]. Se utilizaron sistemas basados en prefijos y en segmentos.

				Prefijos		Segmentos	
Origen	Destino	BLEU[↑]	TER[↓]	WSR[↓]	MAR[↓]	WSR[↓]	MAR[↓]
fr	en	$20,8 \pm 1,0$	$60,7 \pm 1,2$	$52,6 \pm 1,0$	$15,5 \pm 0,3$	$48,3 \pm 0,9$	$23,8 \pm 0,4$
en	fr	$21,0 \pm 0,9$	$61,1 \pm 1,2$	$56,6 \pm 1,0$	$15,4 \pm 0,3$	$51,9 \pm 0,9$	$22,2 \pm 0,3$

respecto del conjunto de test de este mismo corpus. El objetivo de este procedimiento siempre fue poder comparar los resultados de este trabajo con otros estudios en el mismo campo.

Uno de los estudios de referencia al desarrollar nuestro trabajo es el de Domingo, Peris y Casacuberta [10]. En él, se utilizaron algoritmos de traducción estadística para implementar sistemas de traducción interactiva basada tanto en segmentos como en prefijos. En estudios posteriores, se demostró que con redes neuronales se podían mejorar los resultados. Sin embargo, este es uno de los estudios más recientes que combinan IMT basada en segmentos y prefijos. Podemos observar en el cuadro 5.7 cómo nuestros modelos ofrecen traducciones considerablemente mejores. Como ya hemos dicho, la tecnología ha avanzado y el paso a los modelos neuronales ha supuesto una gran mejora. Respecto del sistema basado en prefijos, sí podemos afirmar que los que hemos probado en este trabajo necesitan menos esfuerzo por parte del usuario. por el contrario, los sistemas basados en segmentos propuestos por Domingo, Peris y Casacuberta [10] y los nuestros parecen estar a un nivel notablemente más similar. El salto tecnológico no parece afectar demasiado al sistema basado en segmentos para los pares de lenguas en común con dicho estudio (francés-inglés y alemán-inglés).

Otro de los estudios que se ha tomado como referencia es el realizado por Peris, Domingo y Casacuberta [33]. En él se utilizaron modelos estadísticos y neuronales para desarrollar varios sistemas de traducción automática interactiva basada en prefijos y segmentos. Dichos modelos neuronales estaban basados en la arquitectura propuesta por Bahdanau, Cho y Bengio [3] y fueron entrenados desde el principio sin ningún tipo de preentrenamiento. Nos referiremos solamente a los resultados de estos modelos, mostrados en el cuadro 5.8, por ser la parte innovadora de este estudio. En cuanto a la calidad de las primeras hipótesis, el sistema expuesto en nuestro trabajo parece aportar un BLEU y TER mayores. Recordemos que el BLEU y el TER son métricas a maximizar y minimizar respectivamente. Es por ello que, sin tener acceso a más información, no podemos emitir un juicio claro entre ambos sistemas en este aspecto. En cuanto al esfuerzo de traducción, vamos a comparar por pares los sistemas basados en prefijos y segmentos. Respecto de los primeros, nuestro sistema parece implicar un esfuerzo mucho menor por parte del usuario humano. De hecho, hay una vasta diferencia entre los valores expuestos en el

Tabla 5.9: Extracto de los resultados de los resultados del estudio realizado por Navarro y Casacuberta [27]. Concretamente, mostramos la calidad de las primeras hipótesis y el esfuerzo de traducción (estimado con el mismo método que en este trabajo) de un sistema de IMT basada en prefijos con modelos *mBART*[39] ajustados a cada dirección de traducción.

Origen	Destino	BLEU[↑]	TER[↓]	WSR[↓]	MAR[↓]
de	en	60,5	30,5	36,7	5,8
en	de	64,9	27,9	36,8	5,2
en	es	58,9	33,5	35,3	5,3
en	fr	57,9	40,1	30,7	5,4
es	en	61,1	31,0	36,3	5,8
fr	fr	57,7	34,0	34,7	5,8

artículo de Peris, Domingo y Casacuberta [33] y en este trabajo respecto de los sistemas basados en prefijos. Tal vez, la diferencia se debe al salto tecnológico que supuso la arquitectura *Transformer* [43] y los modelos preentrenados. En lo que se refiere a los sistemas basados en segmentos, el presentado en Peris, Domingo y Casacuberta [33] y el de este trabajo parecen necesitar un esfuerzo por parte del usuario similar. Si bien nuestro sistema necesita un uso mayor del ratón, también presenta un menor uso del teclado en las simulaciones.

Por último, en el estudio de Navarro y Casacuberta [27] se utilizaron grandes modelos de lenguaje en contraposición con modelos neuronales simples. Para ello, se ajustaron modelos de *mBART* para las distintas direcciones de traducción con los resultados expuestos en el cuadro 5.9. Entre otras cosas, se comprobó cómo el realizar un proceso de *fine tuning* a los modelos mejoraba sus prestaciones, también para la traducción interactiva basada en prefijos. Respecto de la calidad de las primeras hipótesis del sistema podemos decir que los modelos entrenados para este trabajo mejoran las del primer estudio. Puede comprobarse este hecho comparando las métricas de el cuadro 5.1 con las mostradas en Navarro y Casacuberta [27]. Respecto del esfuerzo de traducción de Navarro y Casacuberta [27] y nuestro sistema basado en prefijos podemos decir que tienen prestaciones bastante similares. Si bien es cierto que en algunos pares de lenguas nuestro sistema ofrece una ligera mejoría no nos atrevemos a afirmar hasta qué punto es significativa dicha diferencia.

CAPÍTULO 6
Conclusiones

A lo largo de esta memoria hemos explicado cómo se ha implementado un sistema de traducción interactiva con varios grandes modelos de lenguaje. También hemos descrito el procedimiento experimental que hemos seguido para realizar la comparación que teníamos como objetivo y los resultados, fruto de dicha experimentación. Extendiendo el estudio de Navarro y Casacuberta [27], hemos confirmado que los LLM permiten reducir el esfuerzo del usuario en esquemas de traducción interactiva basada tanto en prefijos como en segmentos respecto de estudios previos. Además, hemos ampliado considerablemente la cantidad de modelos y lenguas que se han aplicado a la IMT.

Principalmente en las lenguas con pocos recursos, que constituyen la frontera más activa actualmente en traducción automática, hemos observado comportamientos más interesantes. Existen múltiples variables que influyen en los resultados del sistema, desde el propio modelo base y su entrenamiento hasta la forma de realizar la simulación. Por nuestra parte, desde que hemos obtenido el modelo hasta que hemos realizado la experimentación, hemos mantenido todos los sistemas en igualdad de condiciones para que todos diesen los mejores resultados posibles. De esta forma, todos han sido evaluados conforme a las mismas reglas y utilizando el mismo sistema. Aún así, con la cantidad de grados de libertad existentes sería demasiado reductista atribuir las diferencias en el esfuerzo exigido al usuario por nuestros sistemas a un solo factor. No obstante, prima realizar un análisis de las posibles causas de las citadas diferencias.

Atendiendo primero a los datos del preentrenamiento de los modelos, se seleccionaron aquellos idiomas más anecdóticos en los conjuntos de entrenamiento originales. Así se buscó favorecer una situación incómoda para estos modelos. Esto no ha ocurrido en todos los casos, y los modelos *mBART* y *M2M* han ofrecido muy buenas primeras hipótesis para dichas lenguas. No obstante, con el modelo *NLLB* hemos podido observar que la traducción automática interactiva es una opción recomendable cuando los modelos no ofrecen traducciones lo suficientemente buenas. Realmente, ha sido este modelo el que menos esfuerzo requiere del usuario para estas lenguas cuando se integra en un sistema de traducción interactiva basada en prefijos.

Otra diferencia inherente a los modelos utilizados es su tamaño. El número de parámetros tiende a usarse como un indicio de la potencia del modelo, aunque no sea el único factor importante. En este trabajo, los modelos que menos esfuerzo implican por parte del usuario sí son también los más grandes. En las lenguas con muchos recursos los *mBART* han sido los modelos superiores, mientras que, como ya hemos indicado, para las lenguas de bajos recursos, la mejor opción han sido los *NLLB*. El aumento en el número de parámetros de los modelos a menudo implica un aumento del tiempo de computación necesario. De hecho, podemos ver este fenómeno en la figura 6.1, donde *mBART* es el modelo que más tiempo tarda en producir una nueva hipótesis. No obstan-

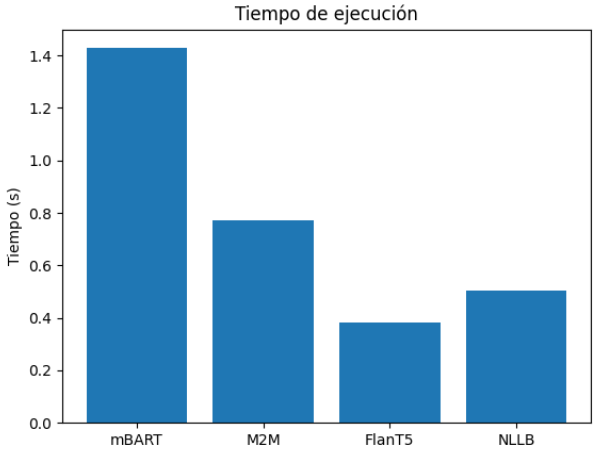


Figura 6.1: Tiempo de computación necesario para generar la siguiente hipótesis con el sistema de IMT basada en segmentos para cada modelo. Las pruebas se han realizado sobre un ordenador equipado con una GPU Nvidia GeForce RTX-4090 y un procesador Intel Core I7 de decimotercera generación.

te, el tamaño tampoco es el único parámetro que influye en el tiempo de computación. Como bien vemos en dicha figura, el modelo *NLLB* utiliza un tiempo mucho menor teniendo prácticamente el mismo número de parámetros. Ello se debe a la particularidad de su arquitectura, que no utiliza todos sus parámetros en una misma inferencia.

Al igual que los parámetros son importantes para medir la potencia de una red neuronal, también su arquitectura es fundamental para realizar bien la tarea para la que se le entrena. En este trabajo todos los modelos se han construido sobre la base de la arquitectura de Vaswani et al. [43], pero todos incluyen alguna diferencia respecto del *Transformer* original. También, respecto de la macroarquitectura existen diferencias, mientras que *mBART* utiliza una aproximación anglocéntrica, el resto de modelos apuestan por un multilingüismo más puro. Así, también la arquitectura es un factor que ha podido influir en los resultados obtenidos por los diferentes sistemas.

Por todo ello, los resultados que obtienen los modelos dependen firmemente de la dirección de traducción sobre la que se apliquen y el protocolo de interacción que se utilice. Principalmente en el apartado 5.2.2 hemos observado que hay modelos que se combinan mejor con esquemas de traducción interactiva por prefijos o por segmentos. Dependiendo de la capacidad del modelo para generar la traducción buscada un paradigma u otro puede necesitar menos esfuerzo del usuario. Como hemos visto en el apartado 5.3, si el modelo necesita ser más guiado, la traducción interactiva basada en segmentos puede ser más efectiva. Por el contrario, si el modelo domina bien la traducción y solo comete pequeños errores, sin duda la traducción interactiva basada en segmentos es la elección adecuada.

Concluyendo, de nuestro estudio deducimos que la traducción automática interactiva basada en prefijos es superior a la basada en segmentos cuando se aplica sobre grandes modelos de lenguaje. La elección de dicho modelo base subyacente ha de hacerse en función de, entre otros factores, los idiomas entre los que busca traducir. Además, dicho modelo base ha de ser ajustado a la dirección de traducción deseada para obtener mejores resultados.

6.1 Trabajo futuro

A pesar de haber realizado una comparación lo más amplia dentro de nuestras posibilidades, hay aspectos en los que no hemos podido profundizar por falta de tiempo. Nosotros hemos realizado nuestro estudio sobre la IMT donde la unidad mínima de las oraciones eran las palabras completas. No obstante, queda abierta la posibilidad de bajar un nivel más y atacar el problema de traducción interactiva a nivel de caracter. Es decir, los segmentos pasan de cubrir palabras enteras a partes de dichas palabras. Mientras que a nivel teórico no supone un gran cambio, a nivel práctico implica tener que alinear los tokens que utilizan los LLMs con las cadenas de caracteres validados. No obstante, esta modificación implica un mayor grado de libertad y versatilidad a la hora de ser aplicado en un entorno real. Como ya hemos abundado en otras secciones, esta flexibilidad es la característica más buscada de los sistemas de traducción basados en posedición.

Además, también nos habría gustado extender la variedad de lenguas que cubre este estudio. Destacamos que las seis lenguas estudiadas utilizan el alfabeto latino, si bien con algunos caracteres propios. Nos gustaría haber estudiado con mayor profundidad cómo el cambio de alfabeto o la dirección de escritura (e.g. sistema árabe) repercuten en las prestaciones ofrecidas por la IMT.

Por último, no hemos podido poner a prueba los modelos más grandes que existen actualmente por las altas necesidades de computación que implican. Pese a haber contado con máquinas más potentes que un ordenador convencional, no hemos podido utilizar los modelos más capaces, dado que son excesivamente costosos de entrenar y utilizar. Es por ello que hemos buscado utilizar los “pequeños” grandes modelos de lenguaje multilingües más representativos de los que hemos tenido conocimiento.

Agradecimientos

Agradecemos a ValgrAI[1] (Valencian Graduate School and Research Network for Artificial Intelligence) y la Generalitat Valenciana[2] su aportación económica a través de las becas a estudiantes del MIARFID (Máster Universitario en Inteligencia Artificial Reconocimiento de Formas e Imagen Digital). Este trabajo también ha sido parcialmente financiado por la Unión Europea (NextGenerationEU/PRTR) bajo el proyecto *FAKE news and HATE speech* (*PDC2022-133118-I00*). También damos las gracias al centro de investigación PRHLT[3] (*Pattern Recognition and Human Language Technology*) por aportar recursos de computación sin los que no hubiera sido posible realizar este trabajo.

[1] https://valgrai.eu/es/
[2] https://www.gva.es/va/inicio/presentacion
[3] https://www.prhlt.upv.es/.

Bibliografía

[1] Vicent Alabau et al. "CASMACAT: A Computer-assisted Translation Workbench". En: *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 2014, págs. 25-28.

[2] Duarte Alves et al. "Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning". En: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, págs. 11127-11148.

[3] Dzmitry Bahdanau, Kyunghyun Cho y Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". En: *CoRR* (2014).

[4] Peter F Brown et al. "The mathematics of statistical machine translation: Parameter estimation". En: *Computational linguistics* 19.2 (1993), págs. 263-311.

[5] Tom Brown et al. "Language models are few-shot learners". En: *Advances in neural information processing systems* 33 (2020), págs. 1877-1901.

[6] Hyung Won Chung et al. "Scaling instruction-finetuned language models". En: *Journal of Machine Learning Research* 25.70 (2024), págs. 1-53.

[7] Seamless Communication et al. "Seamless: Multilingual Expressive and Streaming Speech Translation". En: *ArXiv* (2023).

[8] Marta R Costa-jussà et al. "No language left behind: Scaling human-centered machine translation". En: *arXiv preprint arXiv:2207.04672* (2022).

[9] Ona De Gibert et al. "A New Massive Multilingual Dataset for High-Performance Language Technologies". En: *arXiv preprint arXiv:2403.14009* (2024).

[10] Miguel Domingo, Alvaro Peris y Francisco Casacuberta. "Segment-based interactive-predictive machine translation". En: *Machine Translation* 31 (2017), págs. 163-185.

[11] Maxim Enis y Mark Hopkins. "From LLM to NMT: Advancing Low-Resource Machine Translation with Claude". En: *arXiv preprint arXiv:2404.13813* (2024).

[12] Angela Fan et al. "Beyond english-centric multilingual machine translation". En: *Journal of Machine Learning Research* 22.107 (2021), págs. 1-48.

[13] Albert Gu y Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces". En: *arXiv preprint arXiv:2312.00752* (2023).

[14] Amr Hendy et al. "How good are gpt models at machine translation? a comprehensive evaluation". En: *arXiv preprint arXiv:2302.09210* (2023).

[15] Geoffrey Hinton, Oriol Vinyals y Jeff Dean. "Distilling the knowledge in a neural network". En: *arXiv preprint arXiv:1503.02531* (2015).

[16] Sepp Hochreiter y Jürgen Schmidhuber. "Long Short-Term Memory". En: *Neural Computation* 9.8 (1997), págs. 1735-1780.

[17] Jordan Hoffmann et al. "Training compute-optimal large language models". En: *arXiv preprint arXiv:2203.15556* (2022).

[18] Edward J Hu et al. “LoRA: Low-Rank Adaptation of Large Language Models”. En: *International Conference on Learning Representations*. 2022.

[19] Guoping Huang et al. “Transmart: A practical interactive machine translation system”. En: *arXiv preprint arXiv:2105.13072* (2021).

[20] Albert Q Jiang et al. “Mixtral of experts”. En: *arXiv preprint arXiv:2401.04088* (2024).

[21] Marcin Junczys-Dowmunt et al. “Marian: Fast Neural Machine Translation in C++”. En: *Proceedings of ACL 2018, System Demonstrations*. 2018, págs. 116-121.

[22] James Kirkpatrick et al. “Overcoming catastrophic forgetting in neural networks”. En: *Proceedings of the national academy of sciences* 114.13 (2017), págs. 3521-3526.

[23] Guillaume Klein et al. “OpenNMT: Open-Source Toolkit for Neural Machine Translation”. En: *Proceedings of ACL 2017, System Demonstrations*. 2017, págs. 67-72.

[24] Tom Kocmi et al. “Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet”. En: *Proceedings of the Eighth Conference on Machine Translation*. 2023, págs. 1-42.

[25] Philipp Koehn. “Europarl: A parallel corpus for statistical machine translation”. En: *Proceedings of machine translation summit x: papers*. 2005, págs. 79-86.

[26] Philippe Langlais, George Foster y Guy Lapalme. “TransType: a Computer-Aided Translation Typing System”. En: *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*. 2000.

[27] Ángel Navarro y Francisco Casacuberta. “Exploring Multilingual Pretrained Machine Translation Models for Interactive Translation”. En: *Proceedings of Machine Translation Summit XIX, Vol. 2: Users Track*. 2023, págs. 132-142.

[28] Ángel Navarro, Miguel Domingo y Francisco Casacuberta. “Segment-based Interactive Machine Translation at a Character Level”. En: *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*. 2023, págs. 239-248.

[29] Ángel Navarro Martínez. “Interactive translation with neural models based on the use of mouse actions and confidence measures”. En: (2020).

[30] Myle Ott et al. “fairseq: A Fast, Extensible Toolkit for Sequence Modeling”. En: *Proceedings of NAACL-HLT 2019: Demonstrations*. 2019.

[31] Long Ouyang et al. “Training language models to follow instructions with human feedback”. En: *Advances in neural information processing systems* 35 (2022), págs. 27730-27744.

[32] Kishore Papineni et al. “Bleu: a Method for Automatic Evaluation of Machine Translation”. En: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 2002, págs. 311-318.

[33] Álvaro Peris, Miguel Domingo y Francisco Casacuberta. “Interactive neural machine translation”. En: *Computer Speech & Language* 45 (2017), págs. 201-220.

[34] Matt Post. “A Call for Clarity in Reporting BLEU Scores”. En: *Proceedings of the Third Conference on Machine Translation: Research Papers*. 2018, págs. 186-191.

[35] Alec Radford et al. “Robust speech recognition via large-scale weak supervision”. En: *International Conference on Machine Learning*. PMLR. 2023, págs. 28492-28518.

[36] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. En: *Journal of machine learning research* 21.140 (2020), págs. 1-67.

[37] Stefan Riezler y John T Maxwell III. “On some pitfalls in automatic evaluation and significance testing for MT”. En: *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, págs. 57-64.

[38] Matthew Snover et al. "A study of translation edit rate with targeted human annotation". En: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. 2006, págs. 223-231.

[39] Yuqing Tang et al. "Multilingual translation with extensible multilingual pretraining and finetuning". En: *arXiv preprint arXiv:2008.00401* (2020).

[40] Peter Toma. "Systran as a multilingual machine translation system". En: *Proceedings of the Third European Congress on Information Systems and Networks, overcoming the language barrier*. 1977, págs. 569-581.

[41] Antonio Toral. "Reassessing claims of human parity and super-human performance in machine translation at WMT 2019". En: *arXiv preprint arXiv:2005.05738* (2020).

[42] Hugo Touvron et al. "Llama 2: Open foundation and fine-tuned chat models". En: *arXiv preprint arXiv:2307.09288* (2023).

[43] Ashish Vaswani et al. "Attention is all you need". En: *Advances in neural information processing systems* 30 (2017).

[44] Jiayi Wang et al. "Synslator: An Interactive Machine Translation Tool with Online Learning". En: *Companion Proceedings of the ACM on Web Conference 2024*. 2024, págs. 1023-1026.

[45] Jason Wei et al. "Finetuned Language Models are Zero-Shot Learners". En: *International Conference on Learning Representations*. 2021.

[46] Yuan Yao, Lorenzo Rosasco y Andrea Caponnetto. "On early stopping in gradient descent learning". En: *Constructive Approximation* 26.2 (2007), págs. 289-315.

[47] Changtong Zan et al. "Building Accurate Translation-Tailored LLMs with Language Aware Instruction Tuning". En: *arXiv preprint arXiv:2403.14399* (2024).

[48] Wenhao Zhu et al. "Multilingual machine translation with large language models: Empirical results and analysis". En: *arXiv preprint arXiv:2304.04675* (2023).