Universidad Zaragoza
1542

Trabajo Fin de Carrera

Traductor automático de las lenguas españolas basado en Transformers

Autor

Iván Latre Rodríguez

Tutor

Jorge Llombart Gil

Ponente

Antonio Miguel Artiaga

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2022

AGRADECIMIENTOS

Primero de todo, querría dar las gracias a mi tutora de prácticas Ana Villalba y a mi tutor de TFG Jorge Llombart, quienes me han guiado desde el principio, tanto con este proyecto como con proyectos externos en la empresa. Trabajar con ellos ha sido muy cómodo, y aprender de todo lo que me han enseñado ha sido satisfactorio desde el primer minuto. También, agradecer a Antonio León, director ejecutivo de la empresa en la que se ha desarrollado este proyecto, por concederme esta oportunidad y asignarme un proyecto propio de la empresa.

Agradecer a mi ponente Antonio Miguel, por toda su ayuda brindada durante todos estos meses, tanto para la realización del proyecto, como para mi formación como Ingeniero. También mencionar a Alfonso Ortega, profesor de la EINA, quien también me ha guiado junto con Antonio a lo largo del proyecto.

Por supuesto, también quiero dar las gracias a toda mi familia, principalmente a mis padres y a mi abuelo por el apoyo incondicional en todo momento, por animarme a conseguir lo que realmente quiero y confiar en mí.

Por último, agradecer a mis amigos más cercanos, tanto de la universidad, como de la residencia y como del pueblo, pues ellos también me han acompañado en este camino directa o indirectamente, pero siempre han estado creyendo en mí y animándome.

RESUMEN

Actualmente existe una gran sobrecarga de información en Internet en todos los idiomas, por ello, los sistemas de traducción automática han tomado gran importancia en los últimos tiempos. Dichos sistemas tienen como objetivo principal analizar sintácticamente el texto de una lengua origen y crear una representación transitoria a partir de la cual se genera el texto en el idioma destino. Este proyecto pretende basarse en el desarrollo de traductores automáticos de lenguas de bajos recursos, los cuales se encuentran en plena investigación. Se pretende implementar un modelo que sirva como base para especializar en ámbitos institucionales ó informativos. También se abordarán modelos pre-entrenados con lenguas de altos recursos.
Para la realización de este trabajo se va a hacer uso de técnicas de Deep Learning y Procesamiento del Lenguaje Natural, en especial, utilizando los modelos basados en el Transformer.

ABSTRACT

Nowadays, there is a huge information overload on the Internet in all languages. For this reason, automatic translation systems have recently become. The main objective of these systems is to parse the source language text syntactically and create a transitory representation from which the text in the target language is generated. This project aims to be based on the development of automatic translators of low-resource languages, which are currently being researched. It is intended to implement a model that serves as a base to specialize in institutional or informative fields. Pre-trained models with high-resource languages will also be addressed.
To carry out this work, Deep Learning and Natural Language Processing techniques will be used, especially using models based on the Transformer.

Índice

Capítulo 1

Introducción

Existen infinidad de cosas que debemos agradecer a los avances tecnológicos, en la mayoría de los casos nos facilitan la vida. Una de las tareas que antes de existir una solución tecnológica resultaba bastante compleja es la traducción de textos.

Internet nos brinda la capacidad de acceder a información de todas las lenguas habladas en el mundo, por lo que esto ha agudizado la necesidad de traducir todo tipo de textos. Las herramientas de traducción automática que podemos encontrar nos pueden ayudar y sacar de más de un apuro, sin embargo es muy importante saber como utilizarlas y para que fines, ya que, a pesar de que cada vez son más inteligentes, todavía están muy limitadas con respecto a lo que podemos conseguir con los servicios de un traductor profesional, capaz de ir más allá del simple intercambio de unas palabras por otras.

Por dicho motivo, en los últimos años, se ha incrementado notablemente el interés por los sistemas de traducción automática, lo que implica, a su vez, incrementar exponencialmente el desarrollo de dichos sistemas.

1.1. Motivación

En la era en la que vivimos la mayor parte de la documentación es digital. Los periódicos, revistas, enciclopedias, publicaciones, artículos e incluso los libros se encuentran expuestos, hoy en día, a través de la web. Por dicho motivo, gran parte de la investigación está dirigida la clasificación y análisis de información útil para los traductores.

En la actualidad, gracias a la aparición de Internet es posible acceder a esa información desde cualquier lugar del mundo y a cualquier hora, con solo tener un dispositivo con acceso a este. Muchas veces la búsqueda de información puede verse limitada por el idioma en el que se disponen esos datos. Por ello, a lo largo de los últimos años, muchas tecnologías se han dedicado a crear algoritmos capaces de automatizar

la traducción de la información, tareas que son imprescindible si queremos disponer y entender información de una lengua que desconocemos. Dichos algoritmos consiguen que toda la información del mundo pueda ser comprendida por cualquier persona, sea del país que sea.

En cambio, hay pocos algoritmos que presten realmente atención a lenguas poco habladas o con pocas referencias, pues, por lo general, los traductores automáticos que mejor funcionan son los que se traducen lenguas desarrolladas y muy extendidas en el mundo. Por esto, la investigación se va a centrar principalmente en el estudio y traducción de estas lenguas y cómo adaptarlas a un modelo capaz de traducirlas a la lengua que nos interese.

Este proyecto, por lo tanto, busca conseguir sistemas de traducción automática de calidad. Producir traducciones de calidad es una tarea difícil, pues, para ello es necesario que un algoritmo que tenga representado en el modelo el texto a traducir y, posteriormente, sea capaz de generar una traducción de forma fluida y entendible. Esto implica no sólo tener representadas las palabras que aparecen literalmente en el texto, sino dar sentido a conjuntos de palabras.

Todo lo expuesto justifica la realización de este proyecto, en el que se propone la generación de traductores automáticos aplicados a ámbitos específicos y genéricos. Para ello, se va a hacer uso del Procesamiento del Lenguaje Natural (***NLP***) y de técnicas de ***Deep Learning***.

1.1.1. Descripción del problema

Los sistemas de traducción automática tienen como objetivo principal extraer la información de un texto en un idioma determinado, entenderlo y traducirlo al idioma que se requiera.

Actualmente ya existen técnicas robustas capaces de generar traducciones de textos, y, en muchos casos y sobretodo dependiendo de los pares de idiomas que hayamos elegido podemos encontrar buenos resultados. Sin embargo, hay casos en los que la combinación de idiomas es más compleja y se obtienen peores prestaciones, principalmente en los idiomas de bajos recursos.

Para extraer la información y conseguir traducirla a una lengua objetivo, el significado del texto en el idioma original (origen) se debe restaurar totalmente en el de destino, es decir, en la traducción. Para ello se necesitan amplios conocimientos de gramática, sintaxis (estructura de las oraciones), semántica (significados), etc., de los idiomas de origen y de destino, además de familiaridad con cada región específica. El mayor desafío reside en cómo se pueden producir traducciones de calidad aptas para ser publicadas mediante la traducción automática.

Para ello, se suelen utilizar criterios basados en métodos estadísticos y basados en reglas [1].

- La traducción automática estadística utiliza modelos de traducción estadísticos, cuyos parámetros emanan del análisis de bases de datos monolingües y bilingües. La creación de modelos de traducción estadísticos es un proceso rápido, pero la tecnología depende enormemente de los corpus multilingües existentes. Además, la traducción automática estadística consume mucha CPU y requiere una configuración de hardware amplia para ejecutar los modelos de traducción que permiten obtener niveles de rendimiento promedio

- La traducción automática basada en reglas se basa en incontables reglas lingüísticas integradas y en múltiples diccionarios bilingües para cada par de idiomas. El software analiza sintácticamente el texto y crea una representación transitoria a partir de la cual se genera el texto en el idioma de destino. Este proceso requiere léxicos amplios con información morfológica, sintáctica y semántica, además de grandes conjuntos de reglas. A pesar de esto, a los resultados de la traducción les puede faltar la fluidez que esperan los lectores. En términos de inversión, el ciclo de personalización necesario para llegar al umbral de calidad puede ser largo y costoso.

Por los motivos mencionados, este proyecto propone centrarse en el estudio de traductores automáticos en lenguas de pocos recursos.

1.2. Objetivos del proyecto

El principal objetivo de este proyecto es diseñar traductores automáticos capaces de traducir y transcribir la información de un texto de una lengua a otra, además de la búsqueda de datos útiles para realizar estos traductores, y su correspondiente análisis y limpieza.

Dentro de este objetivo general, se persiguen los siguientes objetivos específicos:

- Búsqueda y análisis de la máxima cantidad de datos posible para lenguas con recursos muy limitados.

- Generar traducciones que no pierdan el contexto de la frase origen, con buena ortografía y signos de puntuación correctos.

- Aplicación de estos sistemas de traducción automática a ámbitos específicos, es decir, se busca realizar modelos genéricos a partir de los que servirán como base para especializar en ámbitos específicos.

- Dimensionamiento correcto que asegure que la memoria necesaria para el sistema de traducción automática no supere nunca a la memoria disponible en la máquina utilizada para la realización del proyecto.

- Evaluación de las traducciones automáticos obtenidas y comparación con traducciones de otros modelos pioneros y las elaboradas por un humano .

- Búsqueda de tiempos de ejecución reducidos. Se pretende conseguir que el traductor necesite el mínimo tiempo posible en traducir.

- Análisis de errores obtenidos y optimización de los mismos para mejoras de calidad.

1.3. Estructura de la memoria

El contenido del presente documento ha sido estructurado tal y como se expone a continuación: · ***Capítulo 1: Introducción***. Presentación del problema abordado, contexto en el que se encuentra y motivación que lo impulsa. También se enumeran los principales objetivos del trabajo. · *Capítulo 2: Estado del Arte*. Funcionamiento y análisis de los traductores automáticos desde sus orígenes hasta la actualidad. Se específican los modelos a utilizar y se analizan con detalle las características de cada uno de ellos. · ***Capítulo 3: Implementación***. Descripción de las etapas o pasos a seguir para la elaboración del algoritmo en cuestión. Se abordarán parámetros con los que se evalúa su calidad ó parámetros que sirven para obtener mejores resultados, una vez analizado el problema.

· ***Capítulo 4: Análisis de resultados***. Se exponen todos los resultados obtenidos con los modelos y la comparación de los mismos. Se muestran tanto los resultados logrados en forma de frases traducidad, como los diferentes parámetros con los que han sido evaluados.

· ***Capítulo 5: Conclusiones***. Conclusiones obtenidas tras la realización del proyecto en cuánto al resultado final y toda su implementación. · ***Capítulo 6: Líneas futuras***. Futura continuación del proyecto, para extender y mejorar su implementación.

Capítulo 2

Estado del arte

2.1. Orígenes e historia de los traductores automáticos

A principios de la década de los años 30 surgió el concepto de traductor automático de la mano de George Artsrouni [2] en Francia y Petr Troyanskii [3] en la URSS. El sistema Artsrouni, se basaba en un sistema de recuperación mecánica automatizada que podía funcionar como un diccionario, pero no pudo llegar lejos a principios de la Segunda Guerra Mundial. El sistema Troyanskii, empezó también como un diccionario automatizado, aunque llegó a implementar componentes electrónicos. A pesar de ello, fue ignorado por el estamento científico soviético.

Una vez terminada la guerra, con el aumento de la guerra fría, la Traducción Automática se convirtió en un tema de interés para las comunidades de inteligencia de ambas superpotencias, tras previamente haberse convertido la criptografía en un tema clave.

En este contexto, el memorándum de Weaver de 1949 sobre la traducción [4] marcó un hito en EE.UU., al defender que la Traducción Automática estaba siendo posible gracias al recién creado ordenador, con una mención incluso a los primeros esfuerzos sobre los perceptrones. Se centró, sobre todo, en la traducción de artículos científicos rusos al inglés. Éstas investigaciones se trataban de algoritmos de maximización de espectativas realizado por humanos, observando reglas gramaticales. También se centraron en la búsqueda de representaciones intermedias entre las frases de origen y las de destino.

A finales de la década de los años 60, el informe ALPAC [5] consolidó el final de la primera fase de la traducción automática, después de haber defendido que la financiación de la investigación estadounidense debería dirigirse a la traducción humana asistida por máquina en lugar de a la traducción automática completa. El sistema intentaba comprender el texto de origen y representar esta comprensión, produciendo

un texto en la lengua de destino a partir de esta representación.

La década de los 80 y los 90 dieron paso a una revolución tecnológica, en la que los ordenadores eran mucho más potentes, sobre todo en cuanto a la capacidad de almacenamiento, pudiendo disponer de bases de datos de texto más grandes. Todo ello acabaría subsumiéndose al Machine Translate (MT) estadístico, a raíz de que el reconocimiento estadístico del habla empezó a dar buenos resultados, gracias a los avances en la teoría de los autómatas y los modelos ocultos de Markov. Estos modelos estadísticos se basaban en el algoritmo de maximización de esperanza ***(Expectation Maximization (EM)*** para aprender tanto los alineamientos entre idiomas -cuáles y cuántas palabras del origen y del destino se corresponden- como un diccionario para traducir después de calcular los alineamientos.

Los primeros modelos neuronales funcionales del lenguaje aparecieron en 2011, 60 años después del memorándum de Weaver, impulsados por redes neuronales recurrentes [6]. La traducción pudo entonces reformularse como una tarea de modelado lingüístico condicional: en lugar de predecir la siguiente palabra más probable, predecir la siguiente palabra más probable condicionada al texto de origen.

En 2016, Google Translate pasó a la MT neural. Los modelos basados en Transformers [7], que eliminan la parte de la red recurrente y solo utilizan módulos de atención iterada, se han convertido en la norma en los últimos años. La potencia de los Transformers se hizo notar rápidamente fuera de la traducción automática y, combinados con el preentrenamiento, forman ahora la columna vertebral de la mayoría de las aplicaciones modernas del Procesado de Lenguaje Natural ***(Natural Language Processing (NLP))***.

Las redes neuronales artificiales y ejemplos de ellas constan en el Anexo C.

Capítulo 3

Solución propuesta

Tras investigar y estudiar la variedad de modelos y estructuras existentes en el mundo de los ***Transformers***, es necesario elegir los modelos óptimos en cada caso. Para ello, se han buscado papers procedentes de investigaciones basadas en traductores, los cuales ofrecen resultados utilizando diferentes modelos. Una vez vistos todos los resultados, se ha concluido en que los mejores han sido obtenidos con el modelo T5, por lo que se ha decidido utilizar dicho modelo para el desarrollo de este proyecto.

También, se ha investigado sobre modelos basados en ***Transformers*** ya pre-entrenados con datos paralelos. Varias universidades del Norte de Europa desarrollaron una serie de modelos, entrenados en cientos de lenguas con bases de datos de Internet. Un grupo de investigación de la Universidad de Helsinki ha realizado varias investigaciones en el campo del ***NLP***, como análisis de datos paralelos para transferir herramientas y recursos lingüísticos a otros idiomas, búsqueda de patrones lingüísticos y la traducción automática estadística.

En este proyecto, nos vamos a centrar, también, en los traductores automáticos de la Universidad de Helsinki, como una alternativa de traductor automático.

Modelo elegido T5 Como bien se ha comentado con anterioridad, para generar la traducción de un texto es necesario escoger un modelo que cuente tanto con codificador como con decodificador, ya que se espera un texto de salida a partir de uno de entrada. Por lo tanto, es necesario elegir un modelo con una arquitectura ***seq2seq***.

El modelo finalmente elegido para la realización de este proyecto es una variante del llamado T5 o ***T***ext to ***T***ext ***T***ransfer ***T***ransformer. Por ello, se va a explicar en detalle su funcionamiento a continuación.

Se le suele denominar un modelo "texto a texto", porque toma un texto como entrada y produce texto nuevo como salida. Este nuevo formato permite aplicar directamente a diferentes tareas el mismo modelo, procedimiento de entrenamiento, función de pérdida, hiperparámetros, codificación... De esta forma, se configura una

sola vez el modelo, aunque es capaz de realizar diferentes tareas. Por ello, es necesario especificar la tarea en cuestión que queremos que se realice, añadiendo un prefijo especifico de la tarea antes de introducir el texto de entrada al modelo. De este modo, el decodificador actúa de una manera u otra en función de la secuencia de salida que se quiere generar, es decir, en función de la tarea a realizar. El codificador, sin embargo, actúa siempre de la misma forma, independientemente de la tarea, debido a que es el encargado de entender la información y el contexto del texto de entrada, necesario en todas las tareas.

Como este modelo es capaz de realizar múltiples tareas hace uso del ***Transfer Learning*** [8] para satisfacerse de ellas. Esta técnica se basa en la reutilización de un modelo previamente entrenado en un nuevo problema, es decir, una máquina explota el conocimiento adquirido en una tarea anterior para mejorar la generalización sobre otra. De este modo, transferimos los pesos que una red a aprendido en una tarea "A" a una nueva tarea "B".

En la figura 3.1 se puede observar un diagrama resumido del marco de trabajo del modelo T5 escogido.

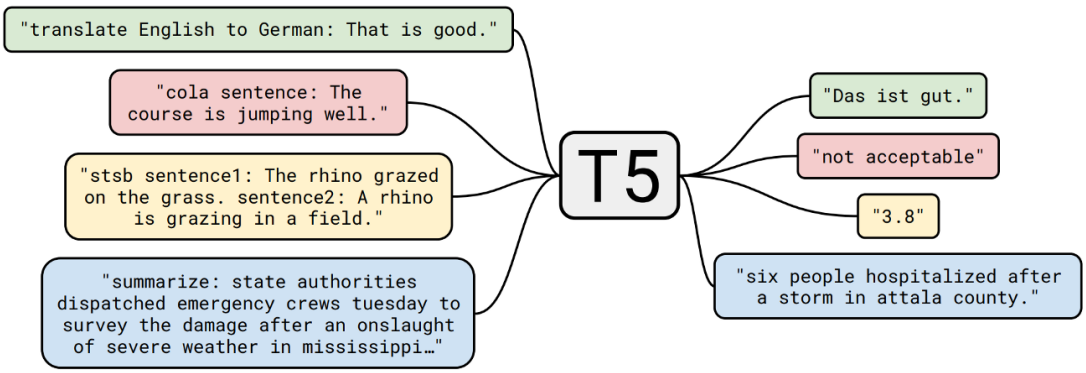

Figura 3.1: Estructura de trabajo del modelo ***T5*** [9]

En primer lugar, este modelo se entrena con texto plano; es decir, necesitamos pares de frases de entrada en el idioma origen y en el idioma destino. Es importante que esta tarea sea rica en datos para que el modelo aprenda la base del idioma en cuestión. Para formarse sobre cómo realizar las tareas que figuran en él, se realiza el llamado ***fine-tuning***, el cual es el encargado de enseñar al modelo pre-entrenado a realizar tareas concretas.

En nuestro caso, para hacer un ***fine-tuning*** de una tarea de traducción, deberemos introducir al modelo un alto número de frases y su respectiva traducción. Una vez realizado el pre-entrenamiento y el ***fine-tuning***, el modelo se considerará capaz de hacer una tarea específica, como, en este caso, traducir.

3.1. Modelo mT5

En el mundo hay aproximadamente 7000 idiomas distintos, de los cuales la mayoría pertenecen a países subdesarrollados. Por lo tanto, los beneficios de la traducción automática no están repartidos equitativamente, sino que se dan por asentados principalmente en países desarrollados. Por el contrario, aún no han llegado a muchos otros usuarios.

El modelo ***T5*** busca realizar tareas en todos los idiomas, beneficiando en especial a los idiomas de bajos recursos. Por esta razón surge una nueva variante del modelo elegido, el ***mT5 (multilingual T5)*** [10] o también llamado ***T5 multilenguaje***.

La característica principal con la que cuenta un mT5 es la llamada transferencia de lenguaje (***Cross Lingual Transfer***). Como hay idiomas dotados con una gran riqueza de datos, las redes neuronales son capaces de aprender mucha información lingüística gracias a ellos. Esta nueva variante del T5 propone transferir esa información aprendida a idiomas que carecen de conocimientos.

El modelo ***mT5*** incluye un codificador, decodificador y bloque de atención, que permanecen sin cambios y se comparten entre todos los idiomas. Este enfoque busca entrenar un único modelo con un conjunto de datos mixto en todos los idiomas. Para organizarlo, se crea una representación compartida donde las palabras con el mismo significado en dos o más idiomas pueden compartir el mismo espacio vectorial. Además, si dos palabras tienen un significado similar se situarán próximas en el espacio vectorial independientemente del idioma al que pertenezcan. De esta forma, se comparte la información para varios idiomas distintos.

Para entenderlo mejor, vamos a poner un sencillo ejemplo. Imaginemos que tenemos un modelo que ha sido entrenado para traducir del inglés-español y español-alemán. El modelo, sorprendentemente, es capaz de traducir inglés-alemán a pesar de no haber sido entrenado en estos idiomas. Igualmente, siempre obtendremos mejores resultados si se realiza un ***fine-tuning*** con las lenguas y dominios específicos.

Por ello, estos modelos consiguen mejorar considerablemente la calidad de tareas en idiomas con pocos recursos. Gracias a esto, se consigue utilizar el procesado del lenguaje natural en todo tipo de idiomas, no solo en aquellos que tienen gran cantidad de datos.

Dentro de esta variante ***mT5***, existen cuatro submodelos dependiendo del número de parámetros internos: ***mT5-small***, ***mT5-base***, ***mT5-large***, ***mT5-XL*** y ***mT5-XXL***. El ***mT5-small*** es el submodelo con el menor número de parámetros, o en otras palabras, el más pequeño. En cambio, el submodelo ***mT5-XXL*** es el más grande de todos. La cantidad de parámetros aumenta progresivamente en el orden en el

que los modelos han sido dados. Esta característica influye también en los requisitos de computación de memoria gráfica, pues cuantos más parámetros contiene el submodelo, mayor memoria ocupa y, en general, mejores resultados obtiene, aunque se necesitarán más datos para entrenarlo. Ocurre exactamente lo mismo, pero al revés: cuantos menos parámetros contiene, menor memoria ocupa y, en general, peores resultados obtiene. Existe ese compromiso entre recursos de computación, datos disponibles y calidad de los resultados.

La memoria es un factor muy importante a la hora de escoger el modelo adecuado, porque por mucho que elijamos el modelo con el que obtendríamos mejores resultados, si el coste computacional es muy elevado, no conseguiríamos nada.

En nuestro caso, la realización de este proyecto se ha desarrollado con un equipo cuya memoria gráfica es de 12 GB. Por lo tanto, el objetivo es encontrar un modelo que se ajuste a la limitación de memoria de la que disponemos. Para saber la memoria exacta que ocupa cada uno de los modelos, hacemos uso de unos ***"BenchMarks"*** que están presentes en la librería ***Transformers***. Estos permiten saber la memoria que ocupa un modelo dependiendo del tamaño de la secuencia de entrada. Para ello, fijamos la secuencia de entrada a un valor de 400 palabras, ya que es el tamaño máximo aproximado que entrará al traductor, y así obtenemos la memoria necesaria para cada unos de los submodelos. Se elige este valor porque en principio no vamos a tener párrafos con mayor número de palabras a la entrada del traductor, y así nos aseguramos de que nuestro modelo quepa en memoria.

Otro parámetro que afecta al consumo de memoria es el ***batch size*** o cantidad de muestras que se propagan a través de la red simultáneamente. Si el valor de este parámetro es menor al número total de muestras de entrenamiento, el entrenamiento se dividirá. A modo de ejemplo, si contamos con 1000 muestras de entrenamiento y configuramos un valor de batch size igual a 100, el algoritmo realizará 10 pasos o ***steps*** de entrenamiento diferentes con 100 muestras nuevas cada uno. Cuanto mayor es la división de este entrenamiento, menos precisa será la estimación del gradiente del mismo, pero menor memoria requerirá. Por el contrario, cuanto mayor es este parámetro, el entrenamiento será mucho más rápido y los resultados mejores.

El consumo de memoria de cada uno de los submodelos se resumen en la tabla 3.1.

Fijándonos en el consumo de memoria gráfica en la tabla 3.1, a priori, cabe la opción de utilizar tanto el submodelo ***mT5-small*** con un ***batch size*** de 2 hasta 4, como el ***mT5-base*** con un ***batch size*** de 2. Aunque nos lo indiquen así los benchmarks, en la práctica no esto es así. El único submodelo que cabe en memoria entrenándose es el ***mT5-small*** con un ***batch size*** de 2, por lo que es el submodelo escogido para la realización del trabajo.

Model	Sequence length (words)	Batch Size	Memory usage (GPU)
mT5-small	400	2	7183
mT5-small	400	3	9593
mT5-small	400	4	11684
mT5-base	400	2	10892
mT5-large	400	2	N/A
mT5-XL	400	2	N/A
mT5-XXL	400	2	N/A

Tabla 3.1: Consumo de memoria gráfica de los submodelos ***mT5***

3.2. Modelos pre-entrenados Helsinki

Una de las alternativas de modelos de traductor automático sustentadas en este proyecto, son los modelos pre-entrenados por el grupo de ***NLP*** de la Universidad de Helsinki. Este grupo de investigación entrenó más de mil modelos de traducción utilizando ***Marian*** [11] con datos paralelos recopilados en ***Opus***, del que hablaremos en la siguiente sección. Más tarde, también convirtieron el modelo entrenado en ***Huggingface Transformers*** y los pusieron a disposición a través de ***Huggingface Hub***, ya que el modelo original estaba desarrollado en C++, a diferencia de Transformers, basados en PyTorch.

El modelo base se trata, como hemos comentado, de ***Marian NMT*** [11], una herramienta de código abierto para entrenar y servir la traducción automática neuronal. Está muy optimizada para traducción. Fue desarrollada, principalmente, en la Universidad de Edimburgo, la Universidad Adam Mickiewicz en Poznań y en Microsoft.

La gran diferencia con los ***Transformers***, es que éstos modelos pre-entrenados usan el mecanismo de atención de ***Bahdanau*** [12]. Éste mecanismo, utilizado antes de que surgieran los ***Transformers***, se emplea para que el decodificador pueda centrarse en los tokens de origen que son relevantes, mientras genera el siguiente token.

A diferencia, los ***Transformers*** utilizan los mecanismos de auto-atención, que ayuda al codificador a codificar la secuencia de forma mucho más eficiente, por este motivo son tan novedosos y revolucionarios.

Dichos modelos pre-entrenados nos van a servir para realizar pruebas y entrenamientos con diferentes idiomas.

Capítulo 4

Implementación

Una vez se ha investigado sobre el estado del arte del procesado del lenguaje natural, en concreto, en el ámbito de los traductores automáticos y se ha elegido el modelo óptimo con el cual realizar el proyecto y también un framework sobre el que apoyarse, es necesario utilizarlo con los datos del dominio específico en el que queremos traducir. Este proyecto se ha dividido en distintas tareas, como podemos observar en la figura 4.1.

Figura 4.1: Diagrama de bloques de la implementación del proyecto

En primer lugar, es necesario realizar una búsqueda del mayor número de datos posible para realizar el entrenamiento del modelo escogido. Posteriormente, se procede a cambiar el formato de los datos obtenidos, así facilitamos su procesado para más adelante. Una vez convertidos al formato deseado, se realiza una normalización y limpieza, con el fin de eliminar las frases erróneas o no válidas, para, así, unir todos

los ficheros procesados en uno único. A continuación, se separa el fichero conjunto normalizado y limpiado en tres fichero distintos, para entrenar y evaluar el modelo en un futuro. Después, el fichero de entrenamiento se convierte al formato que el modelo espera recibir. Por último, se entrena el modelo y se evalúan los resultados obtenidos.

4.1. Búsqueda y clasificación de datos

Para entrenar un modelo y conseguir traducciones coherentes, es necesario disponer de una gran cantidad de datos. Las redes neuronales aprenden a realizar tareas a base de ver ejemplos ya hechos. Por dicho motivo, cuantos más ejemplos hayan visto, más habrán aprendido y mejores resultados serán capaces de obtener. Este es el llamado aprendizaje supervisado.

En este caso, los datos necesarios para entrenar el traductor son pares de frases en los idiomas fuente y destino. Internet, aunque tenga grandes cantidades de datos, muy pocas webs o empresas generan textos traducidos de las webs. Una de ellas se trata de ***opus.nlpl.eu***, como se ha comentado en el capítulo anterior, sustentada por una colaboración de grupos de investigación universitarios en ***NLP***, en el norte de Europa. Esta web, año a año se encarga de traducir bases de datos de las webs en multitud de idiomas y su correspondente alineación.

Está dotada de textos de dominios de distintos ámbitos, tanto genérico como específicos. Por lo tanto, la primera tarea que hay que hacer es descargar y clasificar los datos de la web. Uno de los ámbitos que nos interesan para el traductor es el ámbito genérico, formado por textos simples extraídos de la web. Estos datos van a ser la base de nuestro traductor, a partir de los que conformaremos un primer modelo genérico.

También, dentro de los ámbitos específicos, nos enfocaremos en instituciones (político) e informativos. Sin embargo, nos vamos a encontrar, además, con textos de contenido informático o web, como comandos. Este tipo de datos no nos interesan.

El volumen de datos disponible va a depender principalmente de las lenguas fuente y destino. Para este proyecto, las lenguas abarcadas van a ser el Euskera, Inglés y Español. Se partirá de la base de un traductor Catalán-Español ya realizado en la empresa.

Una de las lenguas más habladas en el mundo, como es el inglés, tiene decenas de millones de pares de frases disponibles. En cambio, en otras como el Catalán y el Euskera, especialmente en éste último, la riqueza de datos es muy limitada.

También, para la búsqueda de textos específicos en Euskera, se ha consultado el Parlamento Vasco, que dispone de una serie de textos traducidos y alineados, además, están en el formato adecuado.

Se puede observar, en la figura 4.2, un resumen de los archivos recolectados de Euskera, tanto de la web ***opus.nlpl.eu*** como del ***Parlamento Vasco***, y su clasificación basada en los dominios que nos interesan:

Nombre	Contenido	Número de frases	ES Tokens	EU Tokens	Formato
WikiMatrix v1	Genérico	0,5 M	381.7M	25.1M	TMX
CCMatrix v1	Genérico	6.6M	91.6M	69.2M	TMX
wikimedia v20210402.tmx	Genérico	38.2k	75.8M	1.8M	TMX
EhuHac v1	Genérico	0.6M	12.0M	9.9M	TMX
Elhuyar v1	Genérico	0.6M	11.2M	8.9M	TMX
EiTB-ParCC v1	Informativos (específico)	0.6M	11.2M	8.6M	TMX
MultiParaCrawl v8	Informativos (específico)	0.5M	10.4M	7.5M	TMX
QED v2.0a	Genérico	18.1k	0.3M	0.2M	TMX
TED2020 v1	Genérico, literatura, religión	10.3k	0.2M	0.2M	TMX
bible-uedin v1	Genérico	7.9k	0.2M	0.1M	TMX
Tatoeba v2021-07-22	Genérico, literatura	2.8k	15.8k	13.7k	TMX
OpenSubtitles v2018	Genérico, subtitulos peliculas	0,8M	11M tokens		TMX
1-2020_LGD_Poliziaren-Legearen-testu-bategina_anonymized	Juridico (especifico)	1,1k	-	-	TMX
9_2017Legea	Juridico (especifico)	5k	-	-	TMX
2011_nazioarteko_hitzarmenak	Juridico (especifico)	6,5k	-	-	TMX
BOE_20210505	Juridico (especifico)	260k	-	-	TMX
BOE	Juridico (especifico)	266k	-	-	TMX
39-2015_40-2015	Juridico (especifico)	3,4k	-	-	TMX
EAE_Legeak_2021	Juridico (especifico)	5,5k	-	-	TMX
3_2021_LO_Eutanasiaren_legea	Juridico (especifico)	0,26k	-	-	TMX

Figura 4.2: Descripción de los datos disponibles Euskera-Español

Otra forma de aumentar el volumen de datos de Euskera específicos se trata del uso del transcriptor de audio disponible en la empresa. A través de éste se recopilaron una gran cantidad de textos transcritos del Parlamento Vasco y de la Televisión Vasca ETB, tanto en Español como en Euskera. La gran ventaja de estos datos es que especializaremos los modelos en los mismos ámbitos que los transcriptores, entrenando con datos que el propio transcriptor ha extraído, por lo que los resultados mejorarán considerablemente.

El problema principal es que no son pares de textos traducidos, sino puras transcripciones del audio. Para poder utilizar estos archivos, hay que conseguir la traducción paralela de éstos, traduciéndolos con otro modelo que sea lo suficientemente potente como para considerar que las traducciones sean válidas. Por lo tanto, para aumentar la capacidad de datos específicos, se investigó sobre modelos de traducción de Euskera para elegir el que mayor prestaciones presentase.

Tras la investigación, se decantó por utilizar el modelo pre-entrenado de ***Helsinki-NLP*** en dirección Español-Euskera. Pero claro, ¿cómo vamos a saber qué tal ha traducido las secuencias si no entendemos Euskera?

Este problema se resolvió utilizando el modelo inverso al que acabamos de mencionar, el ***Helsinki-NLP*** en dirección Euskera-Español. De esta forma, partimos de los textos transcritos en Español, los cuales, tras limpiarse y normalizarse, se traducen con el modelo de ***Helsinki*** en dirección Español-Euskera. Una vez tenemos estos textos traducidos al Euskera, debemos de volver a traducir este archivo con el modelo inverso, que nos devolverá, nuevamente, las secuencias en el idioma origen, en

este caso el Español.

Se realizaron pruebas de traducción con estos modelos pre-entrenados, pequeños ficheros de contenido institucional, para ver si las traducciones eran suficientemente buenas, y los resultados fueron los siguientes:

FRASE 1 ORIGINAL: *Convenio No. 100 relativo a la igualdad de remuneración entre la mano de obra masculina y la mano de obra femenina por un trabajo de igual valor*

FRASE 1 TRAS LAS TRADUCCIONES DE HELSINKI-NLP: *100. Convenio por la realización de un trabajo de igual valor en materia de igualdad salarial entre la mano de obra masculina y las mujeres*

FRASE 2 ORIGINAL: *La esfera jurídica de derechos de los ciudadanos frente a la actuación de las Administraciones Públicas se encuentra protegida a través de una serie de instrumentos tanto de carácter reactivo, entre los que destaca el sistema de recursos administrativos o el control realizado por jueces y tribunales, como preventivo, a través del procedimiento administrativo, que es la expresión clara de que la Administración Pública actúa con sometimiento pleno a la Ley y al Derecho, como reza el artículo 103 de la Constitución.*

FRASE 2 TRAS LAS TRADUCCIONES DE HELSINKI-NLP:
La esfera jurídica de los derechos de los ciudadanos ante la actividad de las Administraciones Públicas está protegida por diversos instrumentos de carácter reactivo, destacando, entre otros, el control realizado por jueces o jueces de recursos administrativos y tribunales a través del procedimiento administrativo, es decir, la expresión clara de que la Administración Pública actúa con plena dependencia de la ley y del derecho, tal y como señala el artículo 103 de la Constitución.

Tras analizar las traducciones anteriores, vemos que el modelo es bastante potente, pues, tras traducir en la dos direcciones, las frases resultantes son muy parecidas a las originales. Es por esto que se decidió utilizar este modelo pre-entrenado para aumentar el tamaño de pares de datos de Euskera.

Por último, se estuvo trabajando justo antes de finalizar este proyecto en la extracción de datos del ***BOE de Euskadi***. Se trata de una web que está dotada de decenas de miles de publicaciones en Euskera con sus respectivas traducciones al Español, desde el año 1936 a la actualidad. La apariencia de la web es la que se muestra en la figura 4.3, en la que diferenciamos la publicación a la izquierda en Euskera y su traducción al Español a la derecha.

Para extraer esta gran cantidad de datos, se automatizó un programa que, tras analizar el contenido de cada URL, conseguía obtener el título, lenguas y contenido de cada una de las publicaciones, exportándolo a un fichero de salida con formato ***JSON***,

EUSKAL HERRIKO AGINTARITZAREN ALDIZKARIA

200. zk., 2018ko urriaren 17a, asteazkena

Laburpenera joan

Bestelako formatuak: PDF (291 KB - 19 orri.) EPUB (213 KB)

Hemen ikusgai dauden gainerako formatuen edukia PDF dokumentu elektroniko ofizial eta jatorrizkoa eraldatuz lortu da

Azterketa dokumentala

XEDAPEN OROKORRAK

INGURUMEN, LURRALDE PLANGINTZA ETA ETXEBIZITZA SAILA

5073

145/2018 DEKRETUA urriaren 9koa, zeinaren bidez behin betiko onartzen baita Araba Erdialdeko eremu funtzionalaren lurralde-plan partzialaren aldaketa, Jundiz-Billodako terminal intermodal logistikoa eraiki ahal izateko.

1.– Araba Erdialdeko Eremu Funtzionalaren Lurralde Plan Partziala behin betiko onartu zen abenduaren 28ko 277/2004 Dekretuaren bidez.

Lurralde Plan Partzial horrek antolamendu integrala ezarri du eremu funtzional osorako, eta udal-mugarte hauek barne hartzen ditu: Alegría-Dulantzi, Armiñón, Arraia-Maeztu, Arrazua-Ubarrundia, Asparrena, Barrundia, Berantevilla, Bernedo, Kanpezu, Zigoitia, Kuartango, Burgu, Iruña Oka, Iruraiz-Gauna, Lagran, Lantaron, Legutio, Otxandio, Urizaharra, Erriberabeitia, Erriberagoitia, Añana, Agurain, Donemiliaga, Ubide, Urkabustaiz, Gaubea, Harana, Vitoria-Gasteiz, Zalduondo, Zambrana, Zuia; baita Badaiako mendilerroa, Entziako partzuergoa, Urizaharra, Lagran eta Laño.

2.– Sustapen Ministerioak eta Euskal Autonomia Erkidegoko Administrazioak lankidetza-protokoloa dute sinatuta Merkatugaien Trenbide Garraioa bultzatzeko Plan Estrategikoaren merkatugaien oinarrizko sarearen barruan dauden terminal logistiko intermodalak garatzeko. Protokolo horren xedea da administrazio bion arteko lankidetza bideratzea Jundiz eta Lezoko terminal logistiko intermodalak garatzeko, zeinak Espainiako Merkatugaien Trenbide Garraioa bultzatzeko Plan Estrategikoaren merkatugaien oinarrizko sarera batu baitira.

3.– 2012ko uztailean, Eusko Jaurlaritzako Lurralde Plangintza, Hirigintza eta Hiri Berroneratze Zuzendaritzak eta Garraio Azpiegituraren Zuzendaritzak elkarrekin sustatu zuten aldaketa honen idazketa; aldaketaren xede nagusia da erabilera logistikoetarako eremu bat gordetzea Vitoria-Gasteizen eta Iruña Okan, Jundiz-Billodako terminal intermodala eraikitzeko. Eremu hori Vitoria-Gasteizko hiri-eremuko mendebaldean dago, eta bat dator egungo tren-trazadurak mugatzen duen eremuarekin –iberiar zabalerakoa–, baita abiadura handiko eta nazioarteko zabalerako etorkizuneko tren-trazadurarekin ere. Eremuak 45,69 hektareako azalera du: horietatik, 28,26 ha Iruña Okan daude, eta 17,43 ha, Vitoria-Gasteizen.

N.º 200, miércoles 17 de octubre de 2018

Ir al sumario

Otros formatos: PDF (334 KB - 21 Pág.) EPUB (216 KB)

El contenido de los otros formatos que aquí se muestran, se ha obtenido mediante una transformación del documento electrónico PDF oficial y auténtico

Análisis documental

DISPOSICIONES GENERALES

DEPARTAMENTO DE MEDIO AMBIENTE, PLANIFICACIÓN TERRITORIAL Y VIVIENDA

5073

DECRETO 145/2018, de 9 de octubre, por el que se aprueba definitivamente la modificación del Plan Territorial Parcial del Área Funcional de Álava Central, para posibilitar la construcción de la Terminal Logística Intermodal de Jundiz-Villodas.

1.– El Plan Territorial Parcial del Área Funcional de Álava Central fue aprobado definitivamente mediante el Decreto 277/2004, de 28 de diciembre.

Este Plan Territorial Parcial establece una ordenación integral para todo el Área Funcional, que comprende los términos municipales de Alegría-Dulantzi, Armiñón, Arraia-Maeztu, Arrazua-Ubarrundia, Asparrena, Barrundia, Berantevilla, Bernedo, Campezo, Zigoitia, Kuartango, Elburgo, Iruña de Oca, Iruraiz-Gauna, Lagrán, Lantarón, Legutio, Otxandio, Peñacerrada-Urizaharra, Ribera Alta, Ribera Baja, Añana, Salvatierra, San Millán, Ubide, Urkabustaiz, Valdegovía, Valle de Arana, Vitoria-Gasteiz, Zalduondo, Zambrana, Zuia e inclusive además los ámbitos de Sierra Brava de Badaya, Parzonería de Entzia, Comunidad de Peñacerrada-Urizaharra, Lagrán y Laño.

2.– El Protocolo de Colaboración entre el Ministerio de Fomento y la Administración Autonómica del País Vasco para el desarrollo de las terminales logísticas intermodales incluidas en la red básica de mercancías del Plan Estratégico para el impulso del Transporte Ferroviario de Mercancías, tiene por objeto instrumentar la cooperación necesaria entre ambas administraciones para el desarrollo de las terminales logísticas intermodales de Jundiz y Lezo, incluidas en la Red Básica de Mercancías del Plan Estratégico para el impulso del Transporte Ferroviario de Mercancías de España.

3.– En julio de 2012 la Dirección de Planificación Territorial, Urbanismo y Regeneración Urbana y la Dirección de Infraestructuras del Transporte, ambas del Gobierno Vasco, promovieron conjuntamente la redacción de la presente modificación, que tiene por objeto principal reservar un área para usos logísticos en los municipios de Vitoria-Gasteiz e Iruña de Oca para la construcción de la terminal intermodal de Jundiz-Villodas. Este ámbito situado al oeste del área urbana de Vitoria-Gasteiz, se corresponde con el área delimitada por el trazado ferroviario actual existente, de Ancho Ibérico y el futuro trazado ferroviario de Alta Velocidad y Ancho Internacional. El área tiene una superficie de 45,69 hectáreas, de las cuales 28,26 corresponden al municipio de Iruña de Oca y 17,43 al de Vitoria-Gasteiz.

Figura 4.3: Vista general de un archivo del ***BOE de Euskadi***

del cual es sencillo extraer los campos.

Estos datos específicos mencionados no se van a contemplar para entrenamiento de modelos en este proyecto, pues hoy en día se siguen limpiando estos datos para realizar una especialización de los modelos genéricos en un futuro próximo.

4.2. Conversión de formato

Los formatos de texto para descargar disponibles son varios, aunque se va a utilizar únicamente la extensión ***.tmx***. La ventaja de esta extensión es que tienen una estructura que facilita el intercambio de memorias de traducción. Las memorias de traducción son almacenes formados por textos originales en una lengua, alineados con su traducción en otras. Así, estos textos, además, están alineados por segmentos, de forma que segmenta el texto en frases tras un signo de puntuación que marca el final de la frase (., ?, !, :, ...) o un salto de párrafo.

A continuación, vemos un ejemplo de este formato de texto, en la figura 4.4:

Como podemos ver, gran parte de la información hay que desecharla, ya que únicamente nos interesan las pares de frases traducidas. Para ello, se analizan los distintos ficheros de texto y se crea un programa en ***Python***. Siguiendo un patrón común a todos los ficheros, obtenemos la información útil, extrayendo únicamente las frases en ambas lenguas. Una vez sacada esa información, se van rellenando ambos archivos de texto, uno por idioma. Podemos observar el resultado tras el cambio de

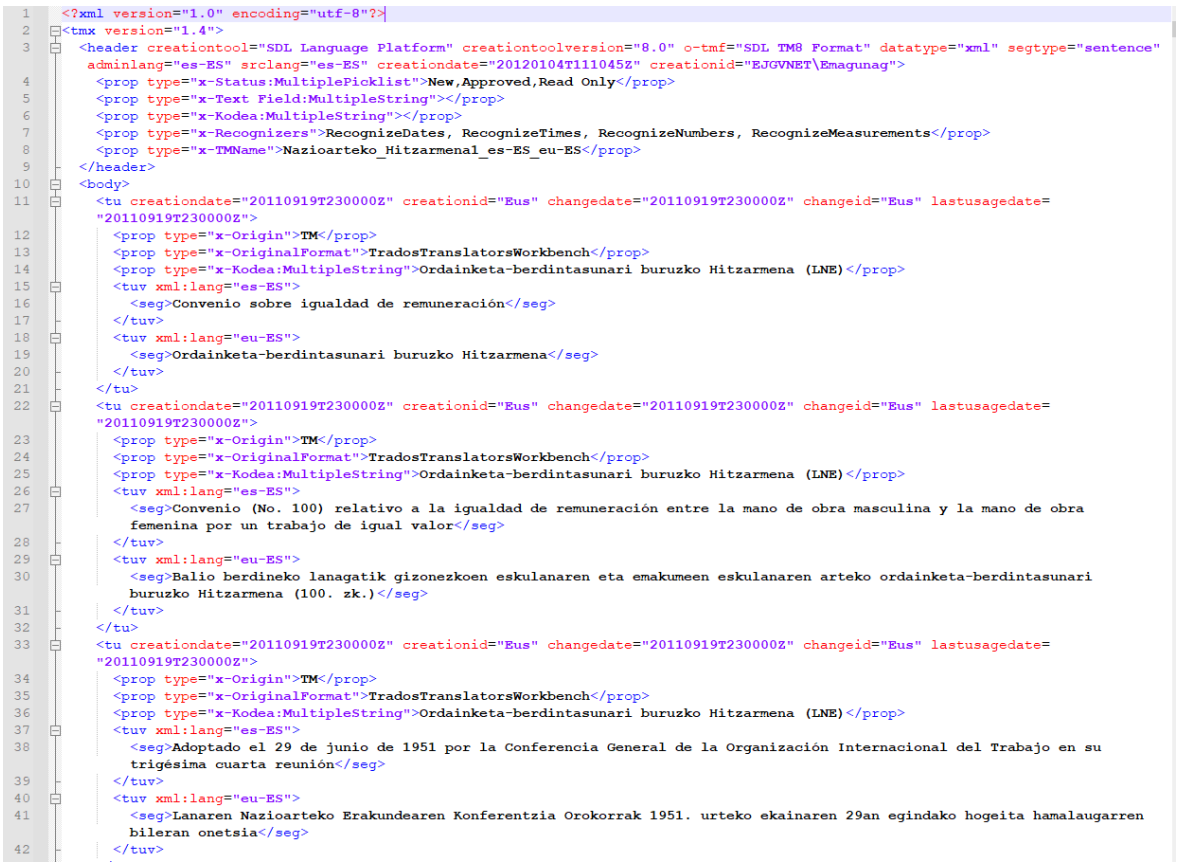

Figura 4.4: Archivo ***tmx*** procedente de la web del Gobierno Vasco

formato del texto de la figura 4.4 en 4.5.

Tras realizar el cambio de formato, conseguimos tener la información mucho más accesible y fácil de manejar, obteniendo una frase por línea, lo que nos implicará una reducción de tiempo notable a la hora de realizar el proyecto.

4.3. Análisis

Analizar estadísticamente los datos es un paso fundamental para saber de qué disponemos. En este caso, la información de interés que se quiere saber es el número de palabras con las que cuenta cada par de frases y su distribución. Conocer esta información sin utilizar un analizador, supondría, nada más y nada menos, que tener que contar la longitud en palabras noticia a noticia y resumen a resumen. Por supuesto el tiempo invertido utilizando esta técnica sería impracticable.

Para ello, se escribe un programa en Python encargado de contabilizar la cifra exacta de palabras que componen cada una de las frases obtenidas.

El análisis de los datos obtenidos es construir con ellos una gráfica. En este aspecto, se ha decidido que la mejor opción va a ser realizar la representación mediante un histograma. De esta forma se verá claramente el tamaño de las frases que disponemos,

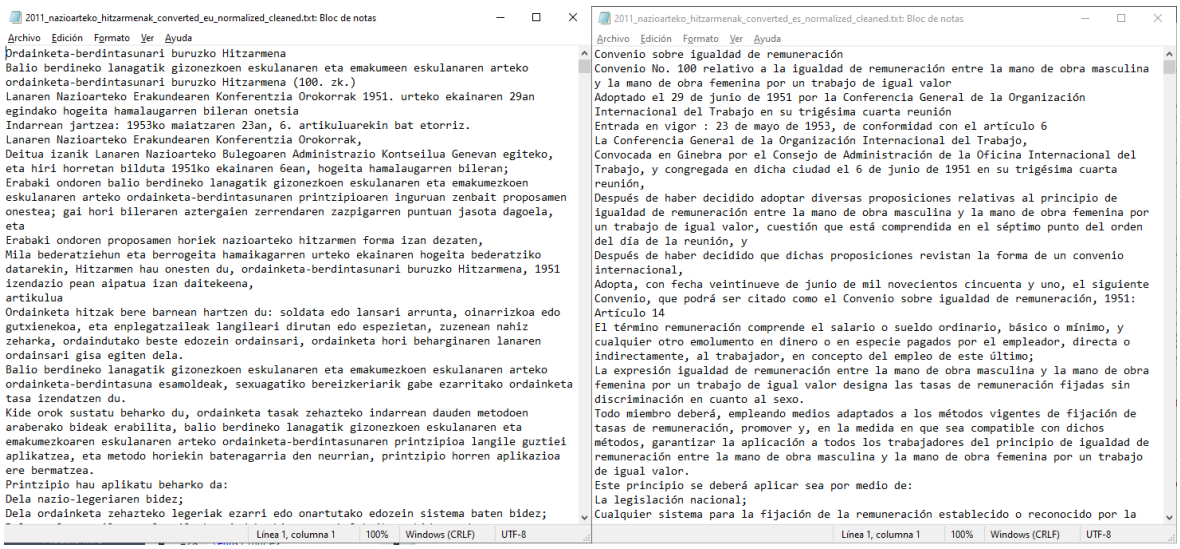

Figura 4.5: Archivo ***tmx*** procedente de la web del Gobierno Vasco tras cambio de formato

en palabras. Podemos ver una representación tras el análisis de los textos obtenidos de contenido genérico en Euskera, en la figura 4.6.

En este análisis se han procesado 90 millones de frases en Euskera, tras haber realizado la conversión al formato adecuado y unirlos en uno único. La gráfica muestra el número de palabras por frase, la cual Nos muestra mucha información sobre los datos que disponemos. Un punto importante a tener en cuenta es que la longitud media de nuestras frases, en este caso, ronda entre 0 y 50 palabras aproximadamente. Esto nos aporta un dato indispensable, mirando con exactitud la gráfica se puede ver que disponemos de miles de frases vacías ó de muy pocas palabras, las cuales no nos van a aportar nada útil a la hora de mejorar el modelo.

También, si realizamos un zoom en la gráfica 4.6, se puede ver que también hay ocurrencias de frases muy largas, en la figura 4.7. Éstas también van a ser descartadas, ya que, para este proyecto, es un tamaño excesivo. ¿Por qué lo consideramos un tamaño excesivo? Como bien explicaremos más adelante, el consumo de memoria del sistema depende proporcionalmente de la longitud de la secuencia de entrada. Para la realización de este proyecto contamos con una memoria limitada y por ello debemos elegir parámetros adecuados que garanticen que el modelo no sobrepase la memoria de la que disponemos. Si introducimos frases con tamaños mucho mayores a la media, pueden aparecer picos de consumo de memoria que sobrepasen la capacidad con la que contamos. Para evitarlo, como bien hemos nombrado anteriormente, eliminaremos las frases con tamaños excesivos.

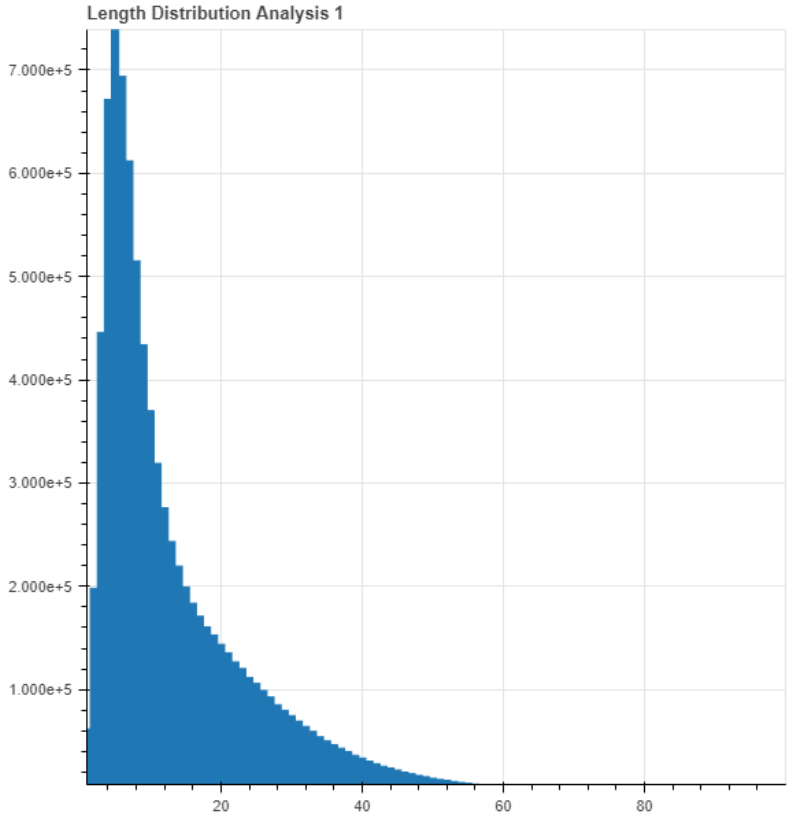

Figura 4.6: Histograma análisis de la longitud de frases de textos genéricos Euskera

4.4. Normalización y limpieza

El proceso de normalización y limpieza se trata de uno de los puntos más importantes previo al entrenamiento, ya que los datos descargados pueden tener errores ortográficos, símbolos duplicados, caracteres fuera del vocabulario, frases en otros idiomas, etc. No corregirlos supondría un gran problema a la hora de desarrollar nuestro traductor, pues, como bien se ha dicho anteriormente, las redes neuronales aprenden en base a los datos que le introducimos. Por lo tanto, si dichos datos son incorrectos, la red aprenderá erróneamente y los resultados no serán buenos.

Nos basamos en el análisis realizado en el punto anterior, de esta forma, tenemos mucha información de nuestros datos antes de realizar la limpieza.

4.4.1. Normalización

El procedimiento de la normalización se inicia con un fichero de configuración, el cual contiene ciertas tareas que se pueden activar o desactivar en función del tipo de normalización que queramos realizar. Un programa principal, escrito en ***Python***, lee este fichero de configuración y carga las tareas escogidas en el ***Normalizador***, en función de la lengua de entrada introducida. No se realiza la misma normalización

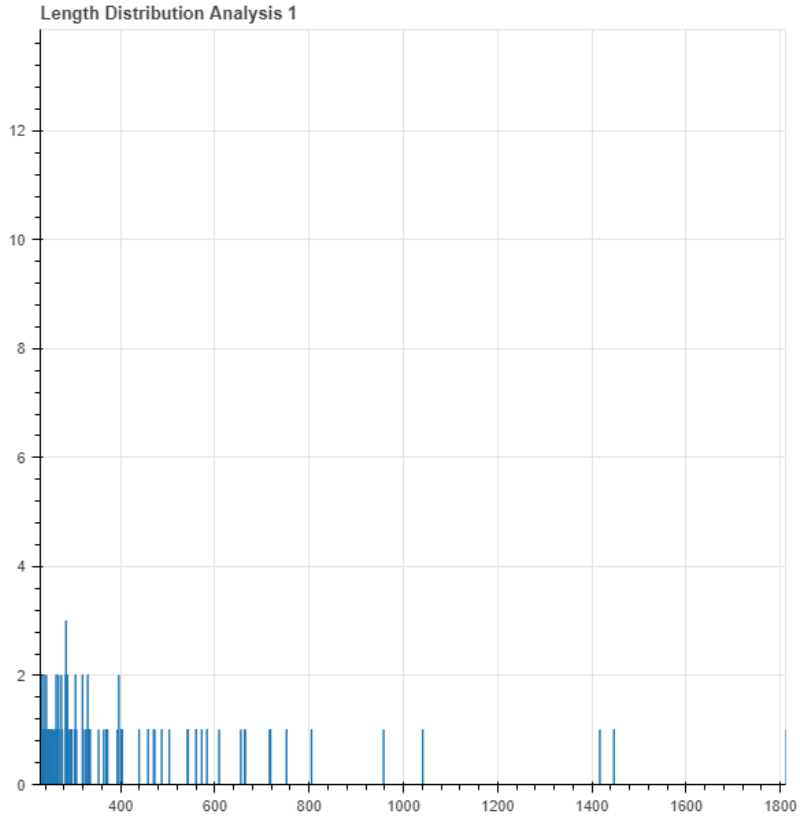

Figura 4.7: Zoom histograma análisis de textos genéricos Euskera

para todas las lenguas, ya que dicho procesado dependen de la ortografía de éstas.

El fichero de configuración del ***Normalizador*** contiene las siguientes tareas:

- Expansión de contracciones. Remplazar acrónimos o abreviaturas por su significado completo ayuda a la red a entender el significado de una frase o texto. Por ejemplo, la palabra **"ayuntamiento"** puede estar presente en varios de los múltiples textos de los que disponemos y por tanto la red la conoce. En cambio, si aparece el acrónimo **"Ayto."** la red difícilmente entenderá el sentido de la palabra. Al transformarla por su significado, la red sabrá que se trata de dos asociaciones.

- Filtrado de espacios en blanco. En la mayoría de textos aparecen varios espacios en blanco o saltos entre frases pertenecientes a tabulaciones horizontales y verticales, avances de página o retornos de carro. Lo único que nos interesa entre frases son saltos de línea y espacios simples, por lo que el resto de separaciones las vamos a eliminar.

- Eliminación de índices. A menudo, en frases pertenecientes al BOE o de temática institucional, la frase comienza con índices, como por ejemplo a), 3.1, -7, entre otros. Necesitamos evitar que el traductor aprenda índices al principio de las palabras, ya que no es relevante a la hora de realizar una buena traducción.

- Unión de varios espacios. La separación entre palabras incluye, a veces, varios espacios en blanco, seguramente por errores a la hora de transcribir los archivos.

Necesitamos unir estos espacios en uno solo, para que la separación entre palabras sea siempre homogénea.

- Conversión de palabras en mayúsculas. Los textos descargados de la web presentan palabras en mayúsculas, principalmente al principio de las frases. Otra tarea de la normalización va a ser convertir estas palabras en mayúsculas a otra en la que únicamente tengamos la primera letra en mayúscula y el resto en minúsculas.

- Eliminación de símbolos fuera del vocabulario. Gran cantidad de frases contienen palabras tanto con símbolos extraños que no pertenecen al vocabulario, como con palabras de otro idioma. A modo de ejemplo, si en un texto en euskera nos encontramos un apóstrofe propio del catalán, un acento o una interrogación al principio, sabemos que éste se tiene que filtrar. Para ello, se define un vocabuario para cada una de las lenguas abarcadas, de forma que, si el símbolo no pertenece al vocabulario preestablecido, el símbolo será filtrado.

4.4.2. Limpieza

Una vez se ha corregido morfológicamente el texto, es necesario realizar una limpieza. Esta limpieza se rige por el uso de **Validadores**. Los procesados principales que realiza el ***Limpiador*** son los que siguen:

- Identificador de lenguaje. Se necesita filtrar las frases que no sean de la lengua requerida, ya que, de lo contrario, el traductor aprenderá pares de frases en lenguas que no pertenecen a la lengua fuente y/o destino. Se ha optado por el uso de un módulo de ***Python*** llamado ***fastText*** que es capaz de reconocer el texto de entrada en función de un umbral predefinido. Si el umbral de acierto está por debajo del preestablecido, el modelo ha detectado que la frase no pertenece a la lengua indicada, por lo tanto es descartada.

- Validador de longitud. A pesar de que el traductor segmente las frases de entrada antes de traducirlas, para el entrenamiento es necesario filtrar frases muy largas para evitar altos consumos de memoria gráfica. De este modo, se establece una longitud mínima y máxima, en palabras. Si estas longitudes no se cumplen, la frase es descartada.

- Validador de frases vacías. Es común en textos descargados que haya frases vacías, las cuales no nos aportan nada, así que éstas se eliminan.

- Eliminación de frases duplicadas. En casi todos los textos disponibles ocurren repeticiones de frases, tanto en los archivos de lengua fuente como en los de lengua origen. Es necesario eliminar las frases duplicadas, ya que no aportan nada útil a la hora de entrenar el modelo. Además, ralentiza el aprendizaje, porque son secuencias que ya ha visto el traductor anteriormente.

Si tomamos como referencia el texto genérico analizado en la figura 4.6, veamos

cómo ha cambiado tras la limpieza, mostrado en la figura 4.8. Ahora, la distribución ha cambiado, de forma que nos hemos quedado con las frases entre 1 y 200 palabras de longitud. Así, nuestro traductor aprenderá a trabajar con este rango de datos, esto no significa que no pueda trabajar con longitudes mayores, pero al trabajar con éstos la calidad será inferior, pues nunca ha visto ejemplos de esa longitud.

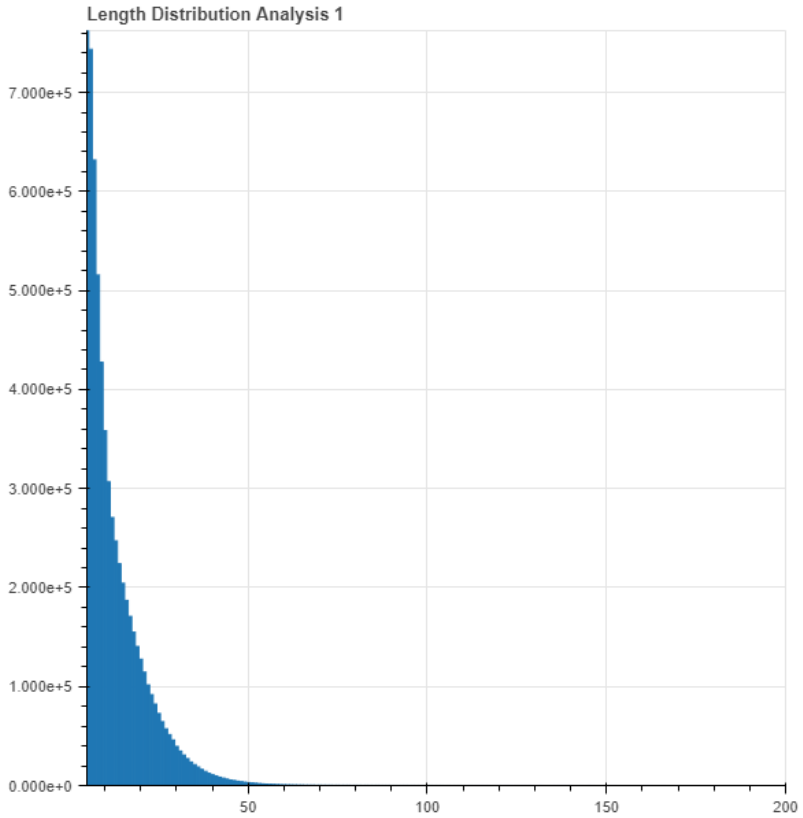

Figura 4.8: Histograma análisis de textos genéricos Euskera tras limpieza

4.5. Split y conversión

Una vez tenemos los datos convertidos, normalizados y limpios, necesitamos seguir transformándolos para que estén en el formato específico a la hora usarlos como entrada a la red neuronal. Para ello, vamos a usar un ***Splitter*** y una última transformación de formato al necesario para el entrenamiento, en ***json***.

4.5.1. Splitter

Disponemos en este momento de los datos listos para entrenar la red, pero, para ello, es necesarios dividirlos en 3 bloques: el bloque de ***train*** (entrenamiento), el bloque de ***dev*** (validación) y el bloque de ***test*** [13].

El entrenamiento de redes neuronales artificiales consta en el ajuste de los pesos y sesgos de las neuronas del modelo, mediante el conjunto de datos perteneciente al bloque de ***train***. Es decir, la red va a aprender a traducir a raíz de ver y procesar secuencias de entrenamiento, por lo que éste bloque va a contener la gran mayoría de los datos disponibles.

Por otro lado, el conjunto de validación o ***dev*** se usa para evaluar el modelo imparcialmente mientras éste se entrena con los datos de entrenamiento.

Usamos estos datos para ajustar los hiperparámetros del modelo. Por lo tanto, el modelo ocasionalmente ve estos datos, pero nunca "aprende" de ellos. Usamos los resultados del conjunto de validación y actualizamos los hiperparámetros de nivel superior. Este conjunto también se conoce como conjunto de desarrollo, ya que ayuda durante la etapa de "desarrollo"del modelo.

Por último, pero no menos importante, tenemos el bloque de ***test***. Este proceso se utiliza una vez que el modelo se considera que está completamente entrenado. Se basa en un conjunto de prueba, es decir, un conjunto de ejemplos que se utilizan únicamente para evaluar el rendimiento del modelo final que ha sido seleccionado durante el proceso de validación. La imagen 4.9 muestra gráficamente el proceso de split de los datos disponibles en sub-bloques, tal y como acabamos de explicar.

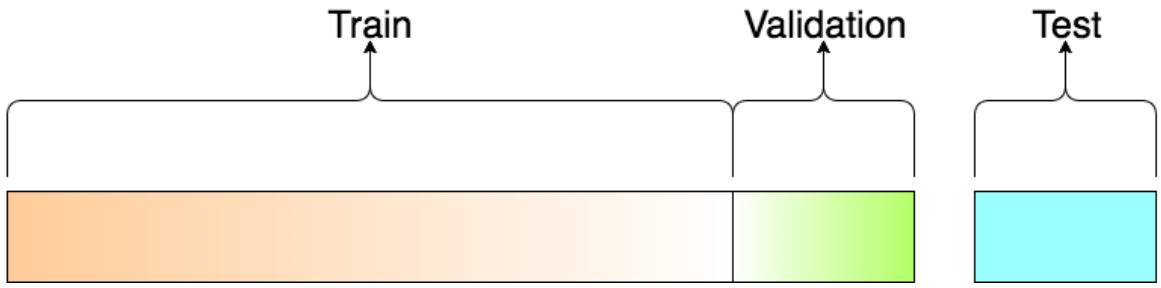

Figura 4.9: División de los datos en sub-bloques [13]

La idea más importante a tener en cuenta es que se necesita una cantidad de datos razonablemente grande para entrenar el modelo (bloque de ***train***) y que así este sea capaz de aprender lo suficiente. En este proyecto contamos con una cantidad de datos razonable para conseguir un modelo preciso, especialmente en cuanto datos de los pares de lenguas Inglés-Español. En cambio, en las lenguas Euskera-Español, los datos disponibles son reducidos a comparación.

Nos vamos a basar en reglas que dicen que, para que se pueda realizar una validación y un test preciso, los bloques de ***dev*** y de ***test*** deben tener, al menos, 50.000 pares de líneas.

Viendo la cantidad de datos de la que disponemos y la información buscada en cuanto a la división, se decide que los datos se van a partir del siguiente modo:

- 99 % de los datos pertenecerán a ***train***. A modo de ejemplo, para los datos genéricos de Euskera, se han conseguido 6,4 millones de pares de frases, y para los específicos de instituciones se ha llegado hasta 272 mil. En cambio, en cuanto a datos específicos de instituciones en inglés, se han obtenido 19 millones de frases.

- 0.5 % de los datos para ***dev***, suponiendo que el tamaño del dataset sea lo suficientemente grande. De lo contrario, deberá aumentarse para tener una validación precisa.

- 0.5 % de los datos para ***test***. Ocurre la misma situación que con los datos de validación.

Así, concluimos con la división de los datos, y sólo necesitaremos un cambio de formato para poder empezar a entrenar los modelos.

4.5.2. Cambio de formato

Antes de realizar el entrenamiento de un modelo, es necesario convertir convertir los textos del lenguaje source y target al formato ***json***, ya que es el soportado por los modelos de la librería de ***Transformers***.

Para ello, como en el resto de procesados, se ha creado un programa escrito en ***Python*** capaz de generar un fichero ***.json*** por cada par de textos con lenguas source y target. Un ejemplo de uso de este tipo de ficheros es el que se muestra en la figura 4.10.

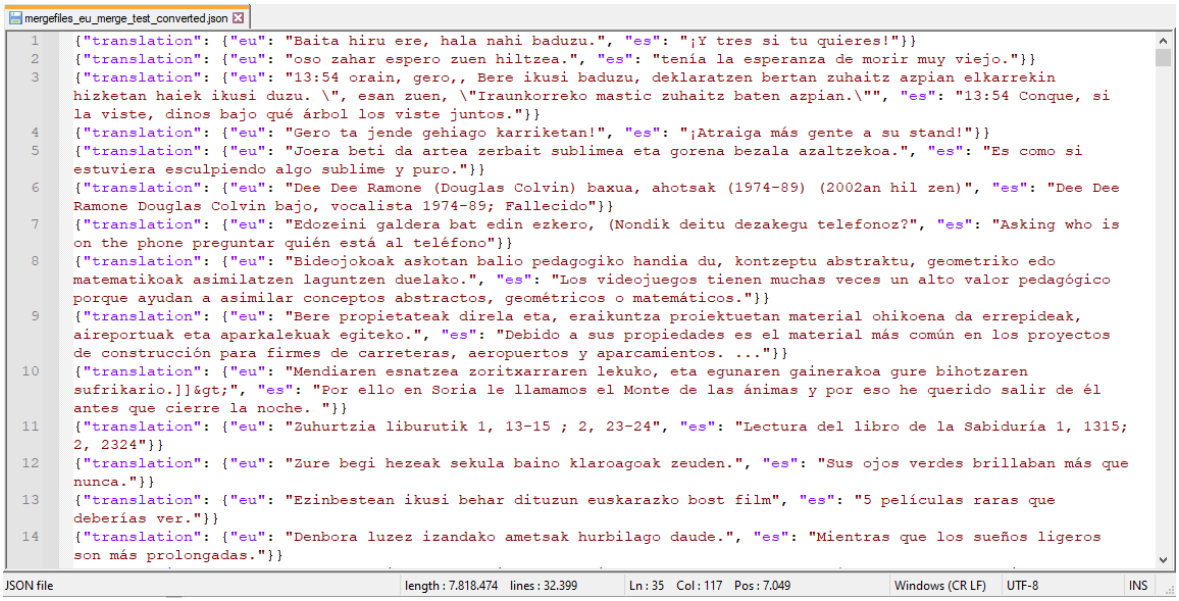

Figura 4.10: Archivo ***.json*** usado para validación

Dicha extensión es una cadena cuyo formato se parece mucho al del objeto ***JavaScript***. Puede incluir los mismos tipos de datos básicos que en un objeto de

JavaScript estándar: cadenas, números, matrices, valores booleanos y otros objetos. Esto le permite construir una jerarquía de datos.

Como se ha observado en la figura 4.10, en nuestro caso, el formato consta de los pares de frases en ambas lenguas, indicada cada una de ellas con su prefijo y precedidas de la tarea que va a realizar nuestro modelo: la traducción.

Tras haber comentado y profundizado en todas las transformaciones de texto, tenemos los datos preparados para entrenar la red neuronal.

4.6. Entrenamiento

El procedimiento para entrenar una red neuronal se resume en un proceso iterativo, a partir del cual se van ajustando paulatinamente los pesos y sesgos (weights and biases) comentados al principio del documento, con el fin de disminuir el sesgo de una función de error determinada.

Se busca que las predicciones de salida sean lo más similar posible a los valores reales, y esto se logra cuando el error es el menor encontrado. La función de error comentada, surge del algoritmo de ***BackPropagation*** [14]. El objetivo principal es obtener un mínimo en la función de error.

El algoritmo de ***BackPropagation*** nos brinda información acerca de cómo deben cambiar los pesos (w) y los sesgos (b), pues se basa en el cálculo de las derivadas parciales de la función de error (o coste) con respecto a cualquier peso o sesgo de la red:

$$\frac{\partial C}{\partial w^L} \qquad \frac{\partial C}{\partial b^L}$$

Figura 4.11: Derivadas parciales ***BackPropagation***

Para encontra dicho mínimo de la función de error, es necesario utilizar un método de optimización. En este caso, el método utilizado es el llamado ***Gradient Descent*** [15], que se trata también de un proceso iterativo a través del cual se busca actualizar los pesos de la red en función del valor del gradiente en dicho punto. Se puede observar gráficamente el método mencionado en la figura 4.12.

Observamos que la función de pérdida tiene su propia curva y gradientes que se pueden usar como guía para ajustar las ponderaciones. La pendiente de la curva de la función de pérdida sirve de guía y señala el valor mínimo. El objetivo es encontrar el mínimo de toda la curva, que representa las entradas donde la red neuronal es más

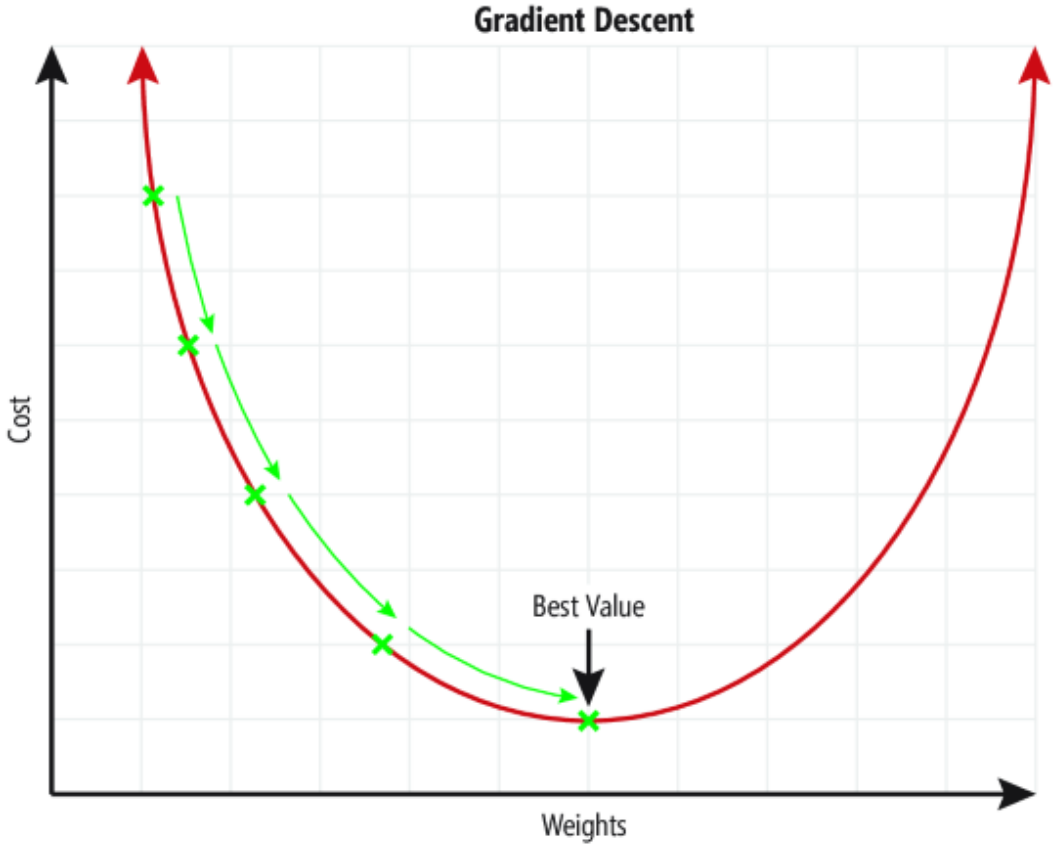

Figura 4.12: Función de error ***Gradient Descent*** [16]

precisa.

Antes de comentar los principales hiperparámetros en los que se basa un entrenamiento, cabe destacar que, si el número de iteraciones de entrenamiento son muy altas, puede que el modelo se sobreajuste a éstos datos y la función de error crezca. A este fenómeno lo denominamos ***overfitting*** [17]. Si esto ocurre, significa que el algoritmo de aprendizaje ha quedado ajustado a unas características muy específicas de los datos de entrenamiento. En otras palabras, el modelo recuerda una gran cantidad de ejemplos en lugar de aprender las características de dichos ejemplos.

Para evitar este fenómeno, se ha optado por usar la técnica llamada ***early-stopping*** [18], cuya idea principal es detener el entrenamiento cuando éste empiece a sobreajustarse. Pero claro, ¿cómo sabemos cuándo es el momento óptimo para detener el entrenamiento? Para ello, volvemos a hacer mención de los datos de ***dev***, que sirven para evaluar el modelo en el transcurso del entrenamiento. Éstos datos nos muestran el error de validación del modelo, o también llamado ***Cross Entropy Loss*** [19]. Dicho error inicialmente disminuye al aumentar las épocas de entrenamiento, pero después al alcanzar un cierto punto, comienza a aumentar o a estabilizarse si existen técnicas

de regularización. Cuando dicho error permanece estable es en el punto en el que el modelo debe detenerse, porque de no ser así, empezará a aumentar de nuevo y, por consiguiente, a sobreajustarse.

Una vez que sabemos ya, en términos generales, el fundamento de un entrenamiento, debemos centrarnos en el dimensionamiento del mismo, es decir, en la elección de los hiperparámetros que queremos que lo caractericen. Los principales para el submodelo ***mT5*** son:

- Longitud máxima de cada secuencia, tanto de source como de target. Viene delimitada por el consumo de memoria, pues secuencias muy largas podrían producir un pico de consumo y, en consecuencia, un ***Out Of Memory (OOM's)***.

- ***Batch size*** del entrenamiento. Es el número de ejemplos que se pasan al algoritmo en cada iteración de aprendizaje, acumulado en lotes o ***batches***. Cuánto más bajo es, menor memoria consume, pero mas lento evalúa y viceversa. En general los batch size mayores provocan entrenamientos con mejores resultados.

- ***Épocas*** de entrenamiento. Se le denomina al número de veces que el dataset de train pasa hacia delante (forward) y hacia detrás (backward) de la red neuronal. En nuestro caso, van a ser siempre valores altos, ya que, para que el modelo se considere entrenado con el método ***early-stopping*** mencionado, los datos de train deben pasar varias veces por la red, de forma que se ajusten los pesos y sesgos hasta conseguir el error mínimo.

- ***Acumulación de gradiente***. La idea detrás de la acumulación de gradientes es, en lugar de calcular los gradientes para todo el batch a la vez, hacerlo en pasos más pequeños. La forma en que lo hacemos es calcular los gradientes iterativamente en batch más pequeños, realizando un recorrido hacia adelante y hacia atrás a través del modelo y acumulando los gradientes en el proceso. Una vez se acumulan suficientes gradientes, ejecutamos el paso de optimización del modelo. De esta forma, podemos aumentar fácilmente el tamaño total del batch a números que nunca cabrían en la memoria de la GPU.

- ***Pasos de evaluación ó Eval steps***. Se trata de cada cuántas iteraciones o steps evaluamos el modelo con los datos de dev. Cuantas más veces evaluemos, dependiendo del tamaño del fichero de dev, ralentizaremos el entrenamiento, pues cada vez que evalúa el modelo no está aprendiendo.

- ***Pasos de guardado ó Save steps***. cada cuántas iteraciones o steps guardamos. Suele ser buena idea evaluar y guardar en el mismo número de iteraciones.

Para calcular equilibradamente Eval steps y Save steps, podemos seguir la siguiente regla:

$$\text{Iteraciones/Epoca} = \frac{TrainSize}{BatchSize_{efectivo}} \tag{4.1}$$

De la que

$$BatchSize_{efectivo} = Batchsize \cdot AcumulaciónGradiente \tag{4.2}$$

Por lo tanto, podemos acabar calculando cada cuántas iteraciones guardamos como:

$$Iteraciones_{validacion}/save = \frac{Iteraciones/Época}{Save/Época} \tag{4.3}$$

Normalmente, podemos guardar y evaluar entre 2 y 4 veces por época, para agilizar el entrenamiento. Para datasets más pequeños, 2 veces por época suele estar bien, pero para datasets más grandes es mejor guardar con más frecuencia. Para visualizar el resultado de estas evaluaciones se monitoriza el entrenamiento a una aplicación externa llamada '***wandb***', donde se exportan los resultados . Ésta nos permitirá visualizar de forma rápida y fácil cómo evoluciona el entrenamiento, resumido en tres pestañas:

- Eval: gráficas referidas a la evaluación del modelo durante el entrenamiento. Podemos ver un ejemplo en la figura 4.13.

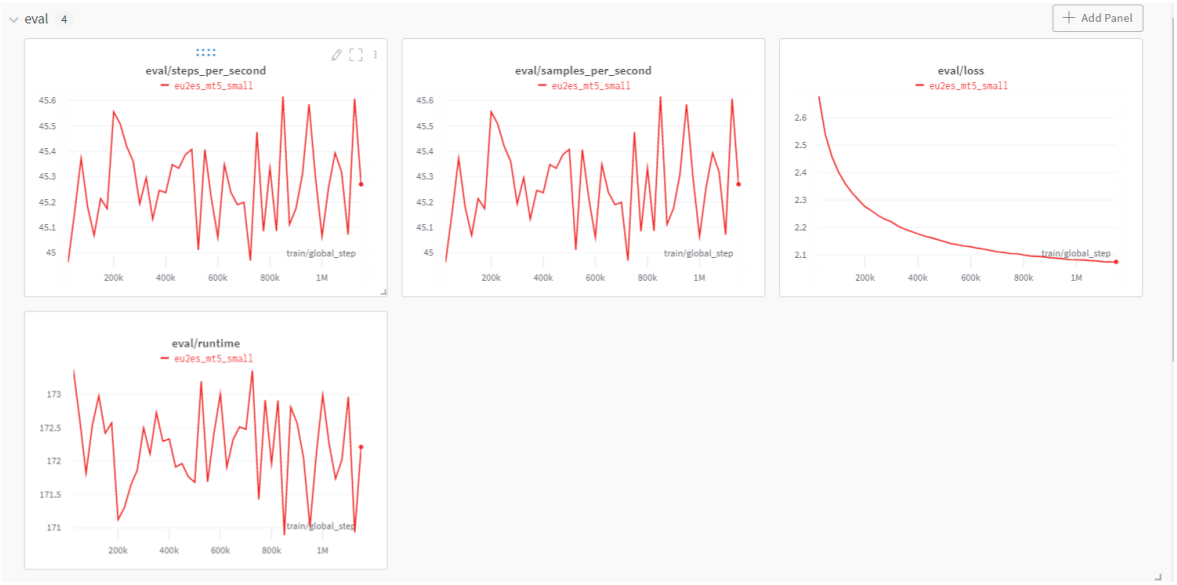

Figura 4.13: Vista general Eval ***wandb***

- Train: gráficas referidas al propio entrenamiento del modelo. Se puede observar en la figura 4.14.

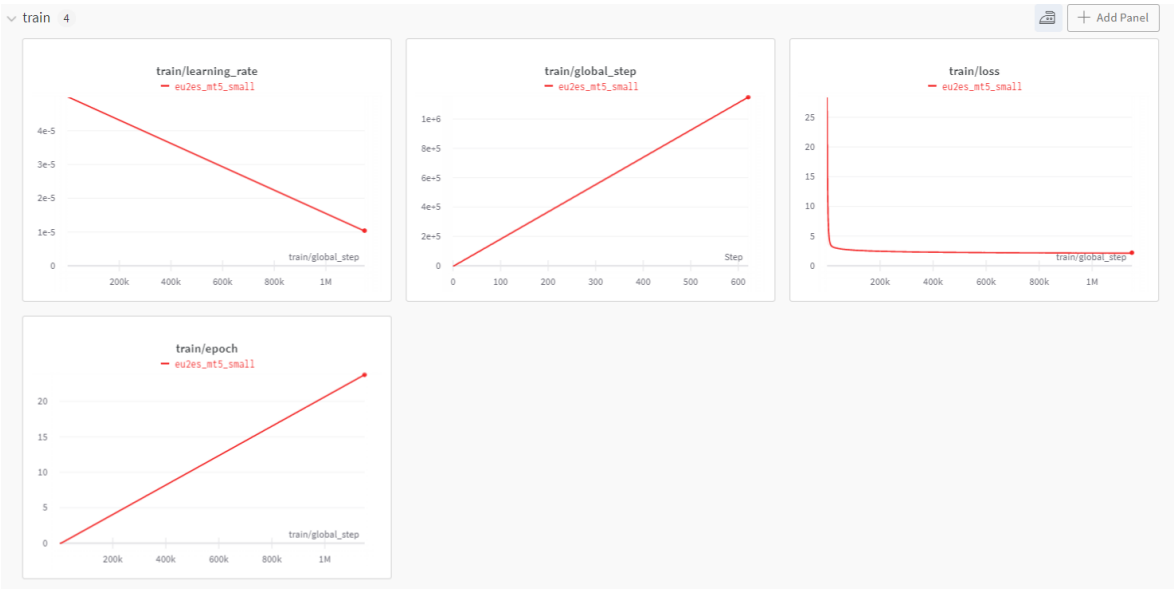

Figura 4.14: Vista general Train ***wandb***

- System: todas las gráficas e información acerca del consumo de GPU y utilización de memoria a lo largo del entrenamiento. La figura 4.15 muestra un ejemplo de este apartado.

Figura 4.15: Vista general System ***wandb***

Una vez se han explicado los conceptos más importantes del entrenamiento y los hiperparámetros que lo dimensionan, a continuación nos centraremos en cada uno de los entrenamientos realizados con sus respectivos modelos. Cabe destacar que los entrenamientos realizados con el modelo de Euskera partieron de la base de un modelo ***mT5*** Catalán-Español, a partir del cual se han realizado los mismos pasos con la lengua

fuente Euskera, tanto para la preparación de datos como para los entrenamientos y la evaluación de los modelos.

4.6.1. Entrenamiento *mT5* Euskera-Español genérico

El primer entrenamiento realizado se basó en el par de idiomas Euskera y Español, centrándose en bases de datos genéricas, a partir de las que, en un futuro, se especializaría el modelo en ámbitos específicos en función de las necesidades y prioridades de la empresa. El modelo entrenado fue el ***mT5***, y se puso como punto de partida el traductor automático Catalán-Español, como se ha comentado anteriormente.

La extracción de datos fue principalmente de la web ***opus.nlpl.eu***, donde se encontraron múltiples bases de datos alineados de dichas lenguas. Se resume en la tabla 4.1 todos los archivos descargados y clasificados de dicha web, preparados para procesar. La tabla muestra el número de frases y tokens en cada lengua de cada uno de los archivos, datos que nos proporciona la web.

Nombre	***Contenido***	***Nº frases***	***ES Tokens***	***EU Tokens***	***Formato***
WikiMatrix v1	General (web)	0,5 M	381.7M	25.1M	TMX
CCMatrix v1	General (web)	6.6M	91.6	69.2M	TMX
wikimedia v20210402	General (web)	38.2k	75.8M	1.8M	TMX
EhuHac v1	General (web)	0.6M	12.0M	9.9M	TMX
Elhuyar v1	General (web)	0.6M	11.2M	8.9M	TMX
EiTB-ParCC v1	Informativos (específico)	0.6M	11.2M	8.6M	TMX
MultiParaCrawl v8	Informativos (específico)	0.5M	10.4M	7.5M	TMX
QED v2.0a	General (web)	18.1k	0.3M	0.2M	TMX
TED2020 v1	General (web), literatura	10.3k	0.2M	0.2M	TMX
bible-uedin v1	General (web)	7.9k	0.2M	0.1M	TMX
Tatoeba v2021-07-22	General (web)	2.8k	15.8k	13.7k	TMX
OpenSubtitles v2018	General (web)	0,8M	11M	11M	TMX

Tabla 4.1: Datos Euskera genéricos

De todos estos archivos, vamos a quedarnos con los que tengan contenido genérico. Tras el procesado, el número total de pares de frases obtenidas para este modelo se resumen en la tabla 4.2 4.2

Nombre	***Tipo de contenido***	***Tamaño original***	***Tamaño Train***	***Tamaño Test***	***Tamaño Val***
Genérico	General (web)	10,27M	6,4M	32,4k	32,4k

Tabla 4.2: Split datos Euskera genérico

Como podemos observar en la tabla 4.2, el tamaño de datos genéricos descargados fueron de 10,27 millones de pares antes de limpiar y normalizar, de los que se obtuvieron únicamente 6,4 millones para train. Destacamos que, para este par de lenguas, el

volumen de datos es muy limitado comparado con el modelo de Catalán, por lo que nos va a marcar grandes diferencias entre estos dos traductores automáticos. Al disponer de este pequeño volumen de frases, se optó por utilizar el 99 % de los datos para ***train***, ya que, sino, obtendríamos peores resultados a la hora de tasar el modelo.

Tras haber comentado los datos de los que partimos y haberlos analizado, podemos definir los hiperparámetros que modelan el entrenamiento, resumidos en la tabla 4.3.

Model	*Batch size*	*Max sequence length*	*Epochs*	*Gradient accumulation*	*Eval steps*	*Save steps*
mT5-small	2	400	30	16	50k	50k

Tabla 4.3: Hiperparámetros del entrenamiento ***mT5*** Euskera genérico

Destacamos el uso de ***batch size = 2*** y de ***gradient accumulation = 16***, limitados por el consumo de memoria, como se ha comentado con anterioridad. Lo mismo ocurre con el hiperparámetro ***max sequence length***, que indica el tamaño máximo en palabras de las frases que entran a la red. El número de épocas se fija a 30, aunque realmente el entrenamiento finalizará siguiendo el criterio de ***early-stopping*** [18]. También puntualizar que el modelo va a evaluar y guardar cada 50k iteraciones, calculado con la ecuación 4.3.

Finalmente, habiendo comentado los hiperparámetros utilizados, veamos el progreso que realizaron los entrenamientso. Nos centramos en el error de evaluación, extraído de ***wandb*** y reflejado en la figura 4.16.

Figura 4.16: Error de validación obtenido durante los entrenamientos del modelo ***Euskera-Español genérico***

Observamos en dicha figura la gráfica del error de validación del entrenamiento

completo frente a las iteraciones, durante un mes y medio aproximadamente. Este fue parado en el momento en el que el error se estabilizo alrededor del valor 1.336 en dos ***eval steps*** consecutivos, si observamos la pendiente de color verde, que fue el último entrenamiento. El modelo fue aprendiendo de los textos de ***train*** hasta que, llegado a ese punto, el error de validación se estabilizó, evitando el fenómeno mencionado de ***overfitting*** [17].

En el transcurso del entrenamiento, se sufrieron varios cortes de luz en la empresa, problema que obligó a volver a lanzar el entrenamiento desde el último checkpoint guardado. Ese es el motivo por el que se distinguen tres curvas en la figura.

4.6.2. Entrenamiento *mT5* Español-Euskera genérico

El segundo entrenamiento realizado se trata del modelo en dirección inversa al del apartado anterior: Español-Euskera, también con la arquitectura ***mT5***. En este caso, los datos de entrenamiento utilizados son los que se han resumido en las tablas 4.1 y 4.2 del modelo anterior. También, los hiperparámetros utilizados fueron los de la tabla 4.3. En cambio, en este caso, el lenguaje fuente es el Español y el objetivo es el Euskera.

El error de evaluación de este entrenamiento se refleja en la figura 4.17.

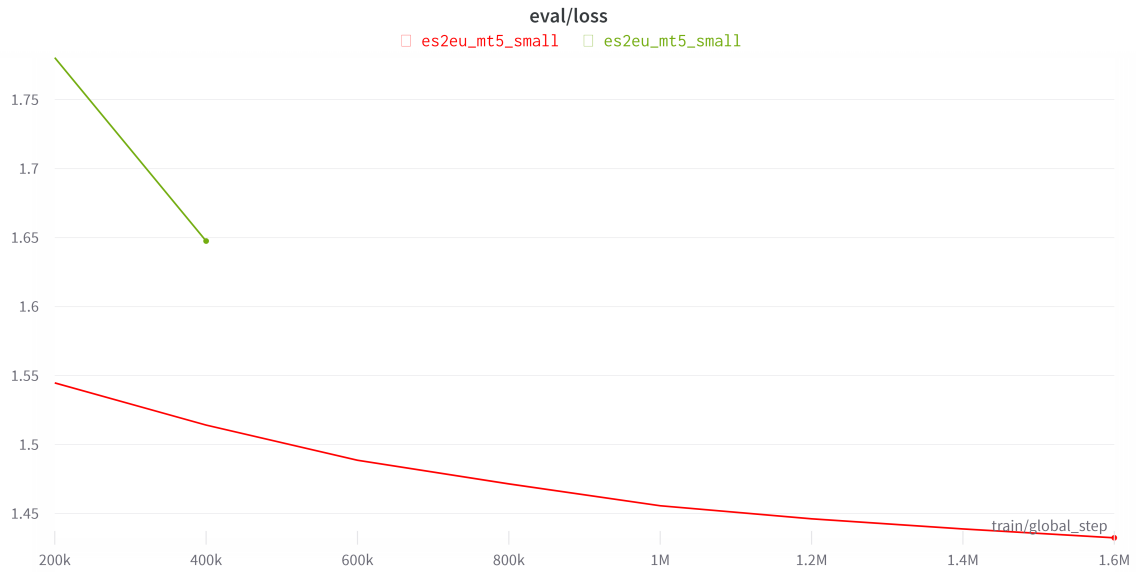

Figura 4.17: Error de validación obtenido durante los entrenamientos del modelo ***Español-Euskera genérico***

Se tuvieron que realizar dos entrenamientos debido a cortes de luz, mismo problema que ocurrió con el modelo previo. La pendiente de error se estabilizó alrededor de 1.43, cuando dejó de disminuir, como podemos ver en la pendiente roja.

4.6.3. Entrenamiento Helsinki Inglés-Español específico

Para la realización del entrenamiento de este modelo, se partió del modelo pre-entrenado ***Helsinki-NLP*** en dirección Inglés-Español. En este caso, éste entrenamiento se ha dividido en distintos sub-entrenamientos, en función del tamaño de datos de ***train***, para evaluar la mejora mientras se va aumentando el volumen de datos paulatinamente.

El Inglés, a diferencia del Euskera, tiene infinidad de bases de datos en Internet, de cualquier ámbito. Por lo tanto, se decidió afinar directamente un modelo específico que partiese de uno pre-entrenado.

La especialización de este modelo se centró en ámbitos ***Institucionales***, como transcripciones de plenos o Parlamento Europeo. En cuanto a la búsqueda de datos, se obtuvieron todos los textos necesarios en la web ***opus.nlpl.eu***, dotada de textos de varios ámbitos específicos.

Como se ha hecho anteriormente, la tabla 4.4 resume todos los archivos clasificados de entorno Institucional, con un tamaño total de frases de 20.2 millones de pares de frases, de las cuales se han obtenido 19 millones para ***train*** y 94 mil para **test** y ***dev***, tras la normalización y limpieza.

Nombre	***Contenido***	***Nº frases***	***EN Tokens***	***ES Tokens***	***Formato***
DGT v2019 v1	Institucional	3.6 M	73.9M	87.4M	TMX
ELITR-ECA v11	Institucional	0.4M	11M	12.4M	TMX
ELRC 2923 v1	Institucional	0.5k	17.9k	20.5k	TMX
ELRC 3382 v1	Institucional	3.7k	84.9k	0.1M	TMX
EUbookshop v2	Institucional	5.3M	220.3M	217.6M	TMX
EUconst v1	Institucional	10.2k	0.2M	87.3k	TMX
Europarl v8	Institucional	1.3M	32.8M	34.2M	TMX
MultiUN v1	Institucional	9.3M	254.4M	300.4M	TMX
News-Commentary v16	Institucional	0.2M	5.4M	6.3M	TMX
UN v20090831	Institucional	74.1k	3.7M	4M	TMX
WMT-News v2019	Institucional	16.6k	0.4M	0.4M	TMX

Tabla 4.4: Datos Inglés Institucionales

Los hiperparámetros utilizados para este entrenamiento, fueron los que refleja la tabla 4.4. Consistió en el desglose del dataset inicial en datasets más pequeños, de forma que el modelo se fuera entrenando desde cero cada vez con más frases, así analizamos la mejora de la traducción en función del número de pares de frases de entrenamiento. También, se decidió realizar únicamente 5 épocas de entrenamiento, para que el experimento no se demorase demasiado.

Tras realizar los primeros entrenamientos, se obtuvo un consumo de GPU muy reducido, por lo que, más adelante, se optó por aumentar el ***batch size*** y el

Model	*Train size*	*Batch size*	*Max sequence length*	*Epochs*	*Gradient accumulation*	*Eval steps*	*Save steps*
Helsinki-NLP-en-es	10k	2	400	5	16	78	78
Helsinki-NLP-en-es	20k	2	400	5	16	156	156
Helsinki-NLP-en-es	40k	2	400	5	16	313	313
Helsinki-NLP-en-es	100k	2	400	5	16	781	781
Helsinki-NLP-en-es	200k	4	400	5	16	781	781
Helsinki-NLP-en-es	400k	8	400	5	16	781	781
Helsinki-NLP-en-es	800k	8	400	5	16	1563	1563
Helsinki-NLP-en-es	1.2M	8	400	5	16	2344	2344
Helsinki-NLP-en-es	2M	8	400	5	16	3906	3906
Helsinki-NLP-en-es	4M	8	400	5	16	7813	7813
Helsinki-NLP-en-es	8M	8	400	5	32	7813	7813
Helsinki-NLP-en-es	12M	8	400	5	32	11719	11719
Helsinki-NLP-en-es	16M	8	400	5	32	15625	15625
Helsinki-NLP-en-es	19M	8	400	5	32	18555	18555

Tabla 4.5: Hiperparámetros del entrenamiento ***Helsinki-NLP*** Inglés-Español Instituciones

gradient accumulation, como se puede ver a partir de 400k pares de train size. Estas modificaciones de los hiperparámetros influyeron muy positivamente, pues el modelo se entrenaba mucho más rápido al haber aumentado el ***batch size efectivo***, incrementando el consumo de GPU, pero nunca llegando a límites de consumo de memoria.

El error de evaluación del entrenamiento de 4M y de 19M se muestra en la gráfica 4.18.

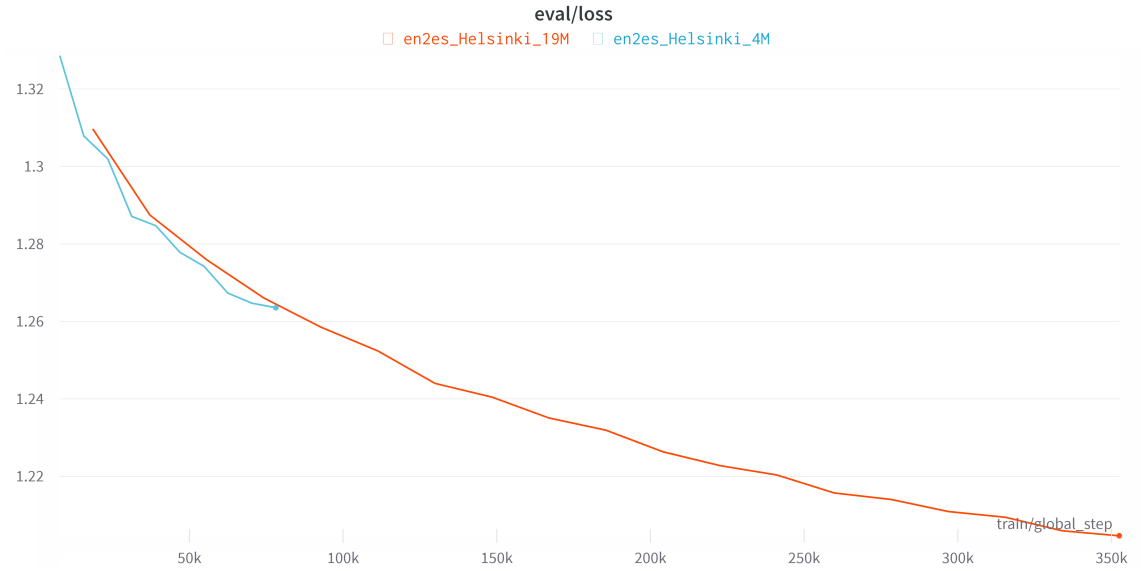

Figura 4.18: Error de validación obtenido durante los entrenamientos del modelo ***Helsinki-NLP*** Inglés-Español Instituciones

Se destaca, como en el resto de los entrenamientos, el descenso de la pendiente de error a lo largo de las iteraciones. En este caso en concreto, el entrenamiento se dió por finalizado una vez se completasen las 5 épocas preestablecidas, por lo que no se hizo uso de la técnica ***early-stopping***. Este experimento cobrará más sentido en el

capítulo de Resultados, donde veremos cuánto ha llegado a mejorar el modelo.

4.6.4. Entrenamiento Helsinki Español-Inglés específico

Por último, concluimos la sección de entrenamientos con el modelo en dirección inversa al anterior. Como ha ocurrido con el modelo inverso de Español-Euskera, los datos de entrenamiento utilizados e hiperparámetros son idénticos.

En este caso, tras haber realizado el experimento en el modelo previo, se optó en este por entrenar directamente el modelo con el train dataset al completo, 19 millones de pares, por optimizar el tiempo disponible. La lengua fuente ahora es el Español y la objetivo el Inglés.

La gráfica del error de evaluación se muestra en la figura 4.19, donde, tras 5 épocas, el modelo seguía aprendiendo paulatinamente, como ocurrió en el caso del modelo Inglés-Español.

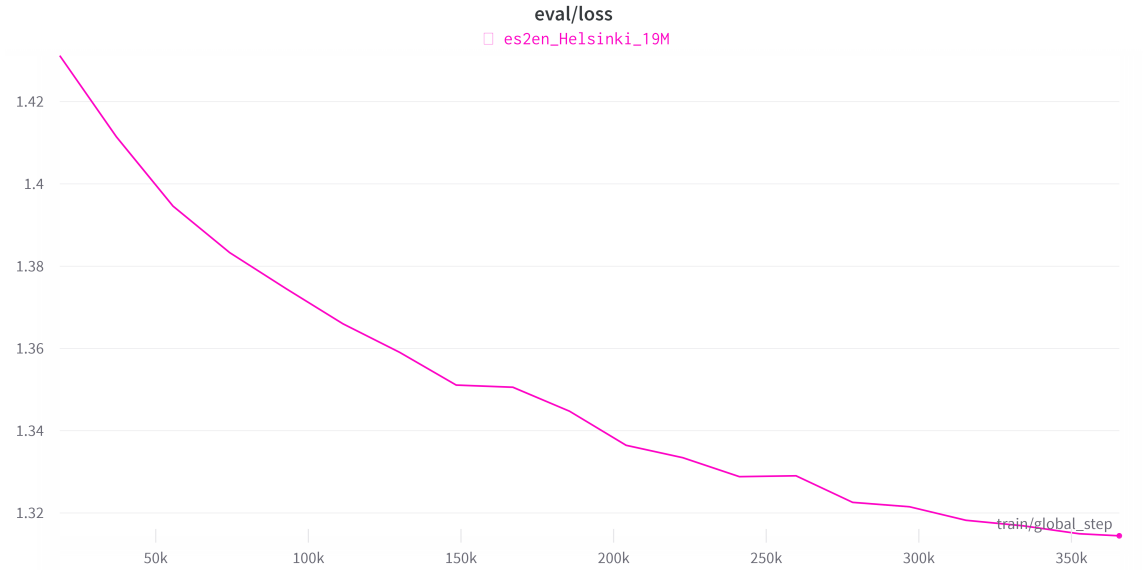

Figura 4.19: Error de validación obtenido durante los entrenamientos del modelo ***Helsinki-NLP*** Español-Inglés Instituciones

Los hiperparámetros utilizados en este entrenamiento se observan resumidamente en la tabla 4.7. El modelo se dimensionó previamente y se calcularon estos parámetros para evitar picos de consumo de memoria. Se entrenaron 30 épocas, aunque se consideró el entrenamiento finalizado bajo el criterio de ***early-stopping*** [18], mencionado en capítulos anteriores.

En cuanto al error de evaluación, visualizamos en la figura 4.20, donde se entrenó hasta que el error se estabilizó en 0.52, aproximadamente.

Model	*Batch size*	*Max sequence length*	*Epochs*	*Gradient accumulation*	*Eval steps*	*Save steps*
mT5-small	2	400	30	16	306k	306k

Tabla 4.6: Hiperparámetros del entrenamiento ***mT5*** Catalán-Español genérico

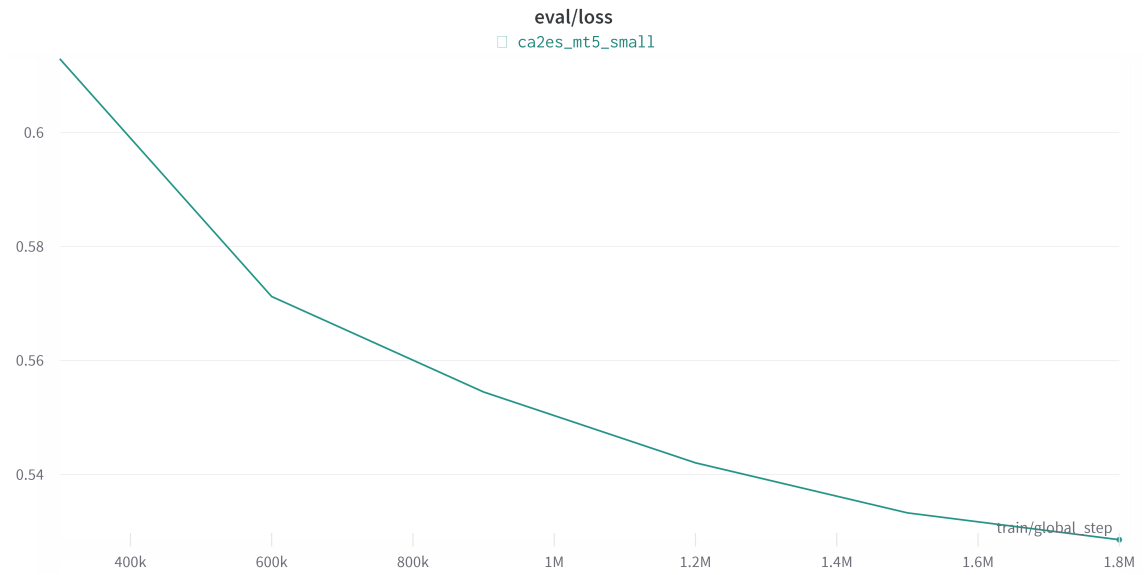

Figura 4.20: Error de validación obtenido durante los entrenamientos del modelo ***mT5*** Catalán-Español Genérico

4.6.5. Entrenamiento *mT5* Catalán-Español genérico

Este entrenamiento fue realizado en la empresa previamente a los entrenamientos con la lengua Euskera, los cuales sirvieron de punto de partida para realizar los traductores Euskera-Español y viceversa. La clasificación de estas bases de datos descargadas de la web, se resumen en la tabla 4.7, donde encontramos el tamaño de los ficheros descargados antes y después de normaliza, además del contenido de cada uno de ellos.

Nombre	*Contenido*	*Tamaño original*	*Tamaño tras limpieza*
Global Voices	Noticias	7k	7k
MultiParaCrawl	Web Crawls, pivotado con inglés	7M	2.5M
MultiCCAligned	Web Crawls, pivotado con inglés	10.6M	1.1M
Wiki Matrix	Wikipedia	1.6M	1.2M
Paracrawl 8.0	Web Crawls	53M	14M
QED	Subtítulos de videos educativos	68k	40k
TED	Charlas de TED	53k	45k
EUBookshop	Colección de publicaciones EU	3k	3k
Tatoeba	Frases cortas genéricas	2.5k	2.5k
OpenSubtitles	Subtítulos de películas	420k	350k

Tabla 4.7: Datos Catalán genéricos

De todos estos conjuntos de datos, se obtuvieron finalmente 19.6 millones de pares de frases para ***train***, 49 mil para ***test*** y 61 mil para ***dev***. Como veremos más adelante, el tamaño de datos de Euskera va a ser mucho más reducido, por lo que los resultados y las métricas de los modelos serán distintas a éste, del que se ha partido de base.

Debemos destacar que el Catalán es una lengua muy similar al Español castellano, por lo que los resultados siempre van a ser mejores si tratamos entre dos idiomas muy parejos. Esto se puede observar en la figura 4.21, donde se superpone la gráfica del error de evaluación de los entrenamientos Euskera-Español frente al entrenamiento Catalán-Español. Deducimos a simple vista, si comparamos la curva lila con el resto, que el error de evaluación del entrenamiento de Catalán es bastante inferior. Esto se debe a lo que se ha comentado en el párrafo anterior.

El volumen de datos de ambos entrenamientos es de 19.6M de pares del Catalán frente a 6.4M de pares del Euskera. En otras palabras, de Euskera disponemos de una tercera parte del volumen de datos del Catalán, esto significa que el modelo Euskera no va a ser tan fuerte como éste. Estos resultados se comentarán más detalladamente en el capítulo siguiente.

Figura 4.21: Error de validación del modelo ***mT5*** Catalán-Español Genérico frente al ***mT5*** Euskera-Español Genérico

4.7. Evaluación

Los entrenamientos finalizado con éxito y como resultado final de los mismos, obtenemos el modelo con un error en validación mínimo. Es el momento de probarlo y evaluarlo en un conjunto de test para asegurar que traduce correctamente.

Además de la calidad de los resultados proporcionados por los modelos, otros dos factores importantes que debemos tomar en consideración a la hora de evaluar el modelo final son:

- ***Tiempo de traducción*** ó tiempo desde que se le introduce una frase hasta que el modelo devuelve su traducción.

- ***Consumo de memoria gráfica*** ó memoria ***GPU*** que necesita el sistema para traducir la entrada dada.

4.7.1. Medición de tiempos y consumos

Para medir el tiempo de ejecución, se ha hecho uso de una librería disponible en Python llamada '***time***'. Se creó un script que realiza medidas temporales, por lo que, usando decoradores alrededor de un punto determinado, será suficiente para medir el tiempo de ejecución del intervalo. Si el uso es iterativo, el script hace una media de cada uno de las medidas temporales obtenidas. Esto lo utilizaremos tanto para medir tiempos de la traducción pura como para el tiempo de carga y de procesado de los modelos, además de medir tareas como el ***Pre-processor*** y ***Post-processor***, dedicados a realizar un procesado antes de introducir la secuencia a la red y tras haber obtenido la traducción.

En cuanto al consumo de memoria gráfica, la herramienta principal usada va a ser las exportaciones de sistema de ***wandb***. También se hará uso de la aplicación llamada '***TechPowerUp GPU-Z***' diseñada para proporcionar información sobre tarjetas gráficas y GPU. Permite medir múltiples parámetros como temperatura, frecuencia del procesador, carga de la GPU, memoria, velocidad del ventilador...

La pantalla principal de la aplicación es la que se muestra en la figura 4.22. Nos muestra la memoria empleada en el campo 'Memory used', y como podemos ver es de 11.6 GB. Como es obvio, en ese momento una red neuronal se estaba entrenando y, por ello, el consumo de memoria era alto, esto es debido a que el entrenamiento es el instante donde se consume mayor GPU.

Otra aplicación para visualizar aún más detalladamente el consumo de GPU y temperatura, entre otras, se trata de las exportaciones del sistema de ***wandb***. En las figuras 4.23 podemos visualizar el porcentaje de ocupación de memoria gráfica a lo largo de uno de los entrenamientos realizados, en función de las horas transcurridas. Se puede ver que el uso es muy alto, debido a que, nuevamente, estamos en un proceso de entrenamiento.

Lo mismo ocurre en la figura 4.24, donde se observa gráficamente en función de las horas de entrenamiento la temperatura de la GPU. Estas herramientas mencionadas nos permiten monitorizar y controlar el entrenamiento, para evitar ***OOM's*** y/o

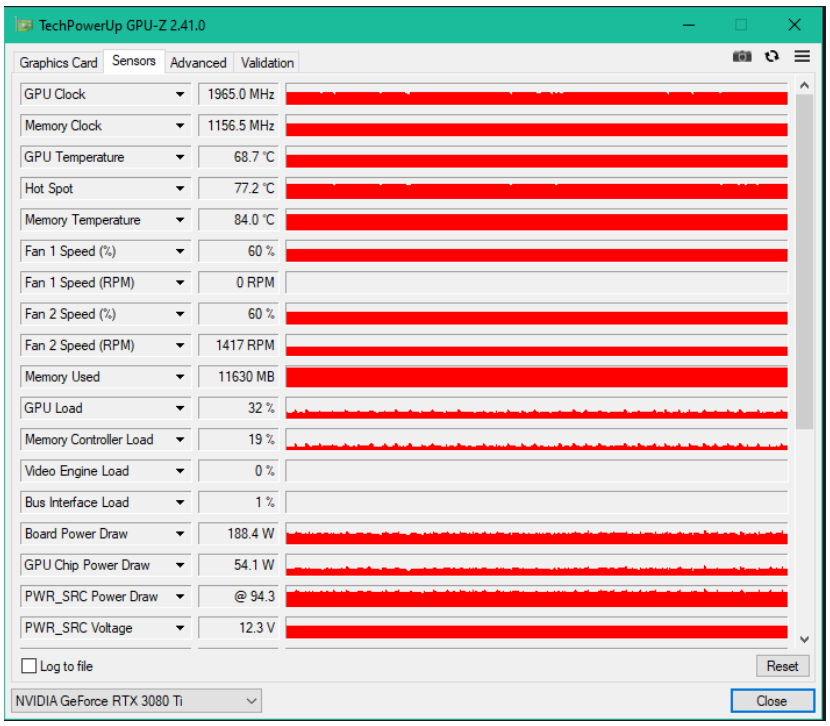

Figura 4.22: Pantalla principal de '***TechPowerUp GPU-Z***' durante entrenamiento

sobrecalentamiento de la tarjeta gráfica.

4.7.2. Métrica utilizada y análisis de hiperparámetros

Llegados al punto de saber cómo medir cada uno de estos dos factores fundamentales, la idea es conseguir un modelo que realice traducciones de calidad, en un tiempo reducido y empleando la menor memoria posible.

Pero, ¿cómo podemos saber la calidad de las traducciones obtenidas? La opción más evidente y fácil sería analizar las traducciones una a una y juzgar con nuestro propio criterio la calidad que éstos poseen. Desafortunadamente, esta elección supondría una pérdida enorme de tiempo y sería cuanto menos factible, aunque también se ha realizado a menor escala.

Investigando sobre distintas maneras de medir la calidad de las traducciones [20], se llegó a la conclusión de que esto fue un problema durante bastante tiempo. Sin embargo, para solventarlo se desarrolló un conjunto de técnicas llamadas ***BLEU score*** [21], comentadas en el Anexo A.

Para tasar cada uno de los modelos entrenados es necesario traducir el fichero de ***test*** generado para cada entrenamiento, pues son datos que la red nunca ha visto, así obtendremos una métrica imparcial de dicho modelo. Esta traducción se comparará con el fichero de test en el lenguaje objetivo (fichero de ***ground truth*** y se generará la métrica correspondiente del modelo.

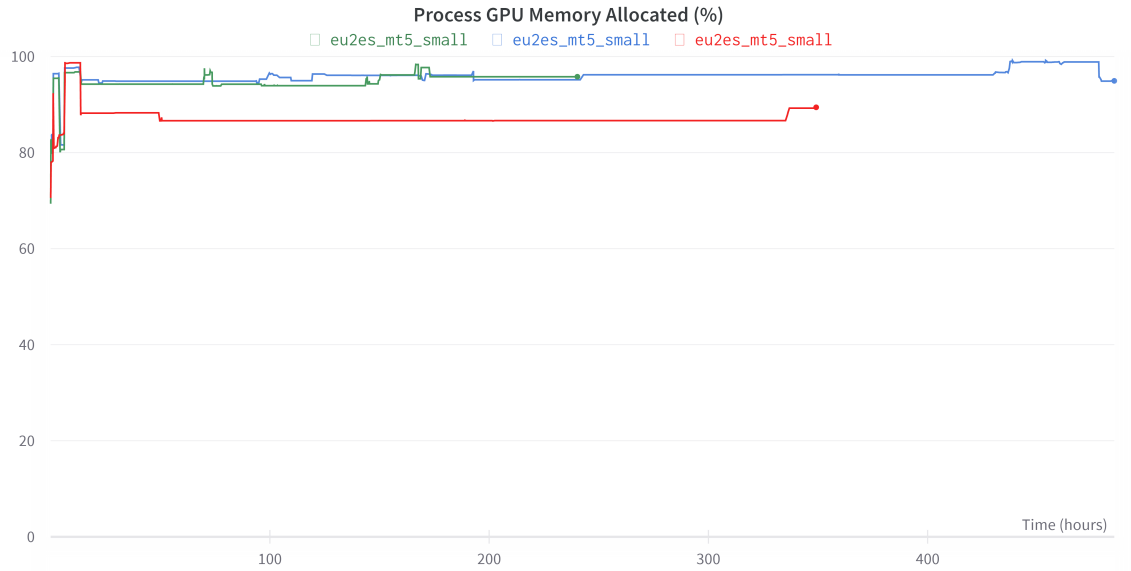

Figura 4.23: Uso de GPU durante uno de los entrenamientos de ***mT5***, con ***wandb***

Antes de esto, es necesario realizar pruebas de traducción, en este caso con el fichero de evaluación (***dev***), para ajustar y estudiar hiperparámetros internos de la red y de la tarea pura de traducción para obtener resultados óptimos. Los principales hiperpaparámetros internos del modelo ***mT5*** con los que se ha experimentado en este proyecto son los siguientes:

- ***Num Beams***. Se basa en el método de decoficicación denominado ***Beam Search*** [22] el cual hace uso de la probabilidad acumulada de las posibles palabras que se van a decodificar. Para entender mejor el funcionamiento de este parámetro, nos remitimos a la figura 4.25, en la que se muestra un ejemplo de decodificación de una secuencia fijando *num_beams=2*.

Así, este parámetro reduce el riesgo de perder secuencias de palabras con mayor probabilidad acumulada, de forma que elegirá la hipótesis que tiene la probabidad acumulada más alta. Refiriéndonos al ejemplo gráfico con el valor fijo de *num_beams=2*, en el segundo paso de decodificación tendríamos por la rama superior la secuencia "The dog has" con mayor probabilidad acumulada, 0.36. Por la rama central, la secuencia más probable en el segundo paso sería "The nice woman", con 0.2. En este ejemplo, el algoritmo ***Beam Search*** va a elegir la secuencia con mayor probabilidad, por lo tanto nos quedaríamos con la primera.

Sin embargo, si el parámetro *num_beams* lo pusiéramos a cero, el resultado cambiaría drásticamente, pues el algoritmo de decodificación escogería, en la primera iteración, la secuencia "The nice...", con una probabilidad de 0.5 frente a 0.4 de "The dog...".

Es un parámetro muy a tener en cuenta, pues es capaz de elegir entre varias

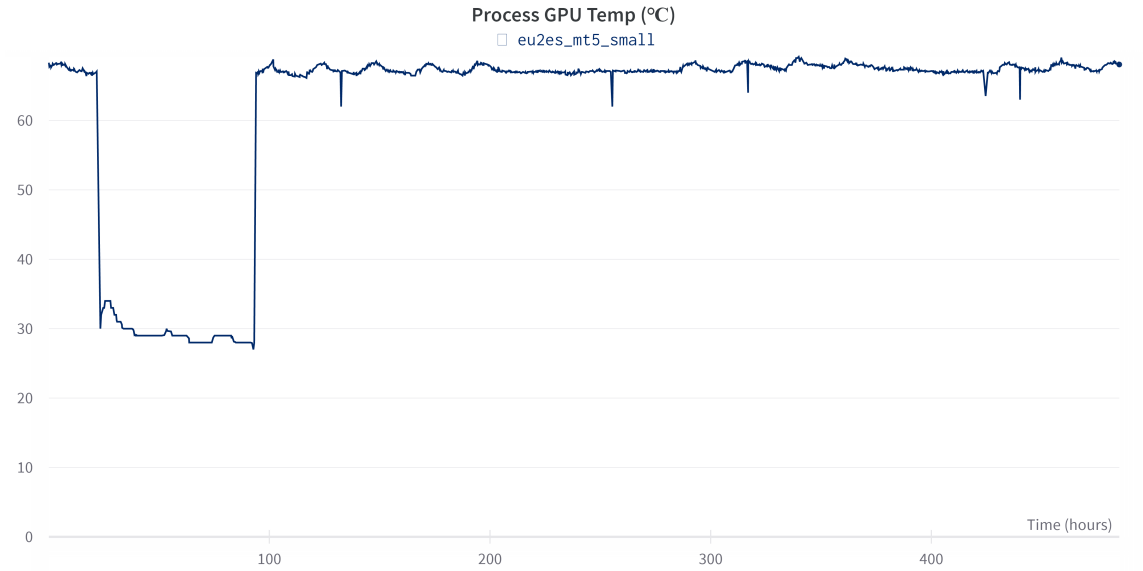

Figura 4.24: Temperatura de la GPU durante uno de los entrenamientos de ***mT5***, con ***wandb***

secuencias bajo la probabilidad acumulada de ellas si lo definimos como mayor de uno, factor que mejorará la calidad de nuestras traducciones.

- ***No Repeat Ngram Size***. En este caso, sirve asegurarse que los ***n-gramas***, explicados en el Anexo A, no se vuelvan a repetir dos veces en la secuencia de salida, configurando manualmente la probabilidad de aparición de éstos a cero.

La idea principal es analizar las traducciones y deducir cuál es el tamaño de palabras que más se repiten al traducir. Va a ser un factor muy importante, pues los ***Transformers***, si el modelo no es lo suficientemente potente, a la hora de traducir vuelve a empezar a decodificar la secuencia en bucle, tras haber acabado de traducir, obteniendo una traducción repetida en varias ocasiones. Es un problema que lleva tiempo estudiándose, y una de las soluciones si ocurre es el uso de este parámetro interno.

Sin embargo, hay que usarlo con cautela. Si, por ejemplo, fijamos su valor a 2 (bigrama) y tenemos una secuencia a traducir que aparezca dos o más veces las palabras "San Sebastián", este parámetro se encarga de poner a cero su probabilidad de aparición por segunda vez consecutiva, por lo que estaríamos eliminando información de la frase original.

Todos estos hiperparámetros comentados presentan también limitaciones de consumo y tiempo, sobre todo ***Num_Beams***, pues, cuanto más grande sea, la traducción será más lenta y consumirá ligeramente más memoria gráfica.

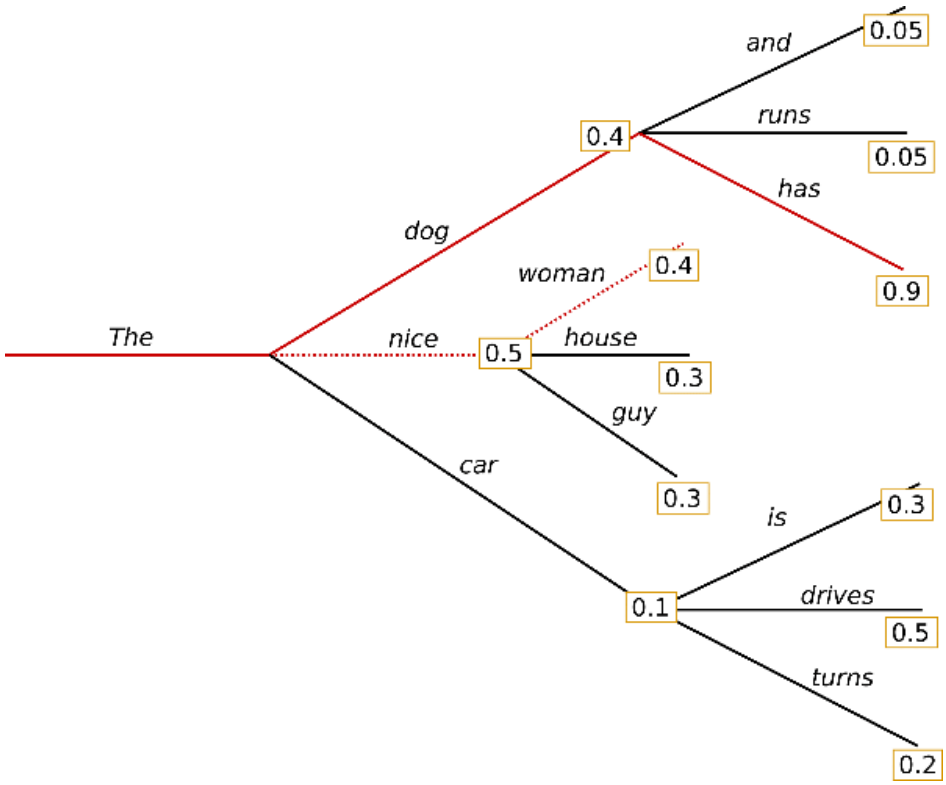

Figura 4.25: Explicación gráfica del parámetro ***num_beams*** [23]

4.7.3. Procesado antes y tras la traducción

Por último, antes de enseñar resultados de traducción, cabe mencionar el procesado al que se someten las frases que se introducen y que salen del traductor. Dividimos estas tareas en tres subtareas: pre-procesado (***Pre-Processor***), ***Core*** del traductor, es decir, la tarea pura de traducción (***Translator Core***) y, por último, un post-procesado (***Post-Processor***).

Pre-Processor

Las secuencias que introducimos al traductor deben de estar previamente normalizadas, es decir, queremos que no sobrepasen el tamaño máximo y que pertenezcan realmente al idioma fuente, además de normalizar las palabras con el mismo criterio que en el entrenamiento. Para ello, se ha desarrollado en ***Python*** una configuración en formato ***.json*** que sea capaz de incluir todos los procesados necesarios, a partir del que un programa principal lo leerá y extraerá la información. Así, tenemos todo el procesado resumido en un fichero intuitivo y fácil de extraer.

El truncado de la secuencia se realiza con una librería interna de ***Python*** abierta, llamada ***spaCy***, diseñada para procesar y entender tamaños grandes de texto, especialmente para el campo de ***NLP***. Esta librería permite segmentar la secuencia de entrada en subsecuencias más pequeñas, si sobrepasan una longitud máxima dada. Si no

se sobrepasa la longitud, simplemente corta por saltos de línea. Puede introducir varias frases a la vez al traductor, o simplemente introducirlas secuencialmente, controlado por un valor de **batchsize** que definirá el número de llamadas al traductor.

La gran ventaja de este módulo es que segmenta las frases con un sentido lógico, tanto por signos de puntuación (“,“, “.“, “...“, “;“, etc) como por conectores entre frases (“y“, “o“)...

Esto es muy importante, porque si las frases no se cortaran con sentido semántico, perdería la lógica de la secuencia y el modelo utilizado no entendería el contexto. Esta librería trunca las frases en función de la lengua dada, ya que cada lengua tiene sus reglas gramaticales propias. Veremos más adelante del proyecto que no hay ningún modelo creado para el Euskera, por lo que habrá que segmentar con otra lengua de entrada.

También, debemos asegurarnos que el idioma fuente es el correcto. Para ello, se ha hecho uso de un módulo interno de ***Python*** llamado ***fastText***, que indica, dado un umbral y la lengua, si la frase de entrada pertenece a esa lengua o no. Es una forma de asegurarnos de que las secuencias de entrada pertenezcan a la lengua fuente. Si no pertenece, transcribe tal cual la frase y no la traduce.

Las secuencias de entrada pueden o no estar normalizadas, ya que, si utilizamos una transcripción de audio de un pleno, por ejemplo, vamos a necesitar una normalización previa, a parte de las subtareas que acabamos de mencionar. Este normalizador es el mismo que se ha usado para normalizar los datos de entrenamiento.

Ahora, ya hemos adecuado el formato de la secuencia de entrada y y está lista para introducirse en la red.

Translation Core

Entramos en la tarea pura de traducción, que consta del ***Core*** interno del traductor. En esta subsección cabe destacar varios hiperparámetros: la longitud máxima y mínima de la secuencia de salida generada y la anchura de haz (ó ***Beam_width***).

Este hiperparámetro es otro factor importante para la traducción. Fijándolo mayor que uno, el decodificador será capaz de decidir entre más secuencias a predecir en función de su probabilidad. Si observamos el ejemplo de la figura 4.25, el número de palabras a decidir para la siguiente predicción se define con esta variable. Cuanto mayor sea, tendremos un abanico de posibilidades de decidificación mayor, a partir de las cuales el algoritmo escogerá la que tenga mayor probabilidad.

Post-Processor

La secuencia ya ha pasado por la red neuronal y hemos obtenido su traducción en el idioma requerido. Por último, hay que realizar un procesado tras esta traducción, pues, como se ha comentado, los modelos entrenados de ***Transformers*** que no son lo suficientemente potentes, el decodificador entra en bucle y saca una traducción con repeticiones.

Para solucionar este problema, se propuso escribir un script en ***Python*** que limpiara manualmente estas repeticiones, si las hubiese.

Una vez hemos comentado todos los hiperparámetros y procesados necesarios para realizar traducciones, pasamos a explicar y analizar los resultados que se han obtenido durante varias pruebas de traducción con los modelos entrenados.

Capítulo 5

Análisis de resultados

En este capítulo se van a enseñar los resultados obtenidos. A lo largo de este proyecto, se han realizado múltiples entrenamientos y pruebas de traducción para intentar optimizar al máximo el resultado final. En este apartado se mostrará varios de estos ejemplos.

Para los entrenamientos de los pares Euskera-Español y viceversa, como bien se ha mencionado a mediados del documento, se ha utilizado el modelo ***mT5-small*** en su implementación en el framework Transformers, siguiendo los mismos pasos que el traductor Catalán-Español. En cambio, también se ha experimentado con los modelos pre-entrenados ***Helsinki-NLP*** para entrenar un traductor Inglés-Español y viceversa. Además, se ha hecho uso de una memoria gráfica ***NVIDIA RTX 3080*** de 12 GB de memoria GPU. El tiempo total de entrenamiento fue de unos 40 días por modelo entrenado para el caso de los ***mT5***. Para los entrenamientos con ***Helsinki-NLP***, los de menor volumen fueron cuestión de horas y los de mayores volúmens de datos, varios días.

Para evaluar los resultados que son capaces de obtener cada uno de los modelos entrenados, es necesario iniciar el periodo de prueba. Es ahora cuando entra en juego el bloque de ***test***. Se introducen todos los datos pertenecientes a ***test*** como pares de frases al traductor y se generan sus respectivas traducciones. Con estas traducciones y las ya existentes en el idioma objetivo, se evalúan los resultados y se obtienen las métricas definitivas de los modelos.

Comenzamos con la evaluación de cada uno de los modelos con el análisis de calidad. Para ello, calcularemos las diferentes ***BLEUs*** a partir del fichero de ***test*** de cada uno de los modelos, de las que podremos estimar su calidad de traducción. Los resultados obtenidos para todos los modelos ***mT5*** definitivos que abarca este proyecto se resumen en la tabla 5.1, donde se han obtenido la métrica ***BLEU*** para los dominios de informativos y genérico. También observamos los resultados que se obtuvieron en el catalán previamente en la tabla 5.2.

Source language	*Target language*	*Modelo*	*Test set*	*BLEU (4-gram)*	*Mod. Pred. Scores (unigram/bigram/trigram/4gram)*
Euskera	Español	mT5 Genérico	Genérico	20,64	43.0/25.5/16.2/10.2
Euskera	Español	mT5 Genérico	Informativos	24,15	47.7/29.6/19.3/12.5
Español	Euskera	mT5 Genérico	Genérico	14,5	38.5/19.1/10.3/5.8
Español	Euskera	mT5 Genérico	Informativos	5,43	21.3/7.5/3.4/1.6

Tabla 5.1: ***BLEUs* obtenidas con los modelos de Euskera *mT5***

Source language	*Target language*	*Modelo*	*Test set*	*BLEU (4-gram)*	*Mod. Pred. Scores (unigram/bigram/trigram/4gram)*
Catalán	Español	mT5 Genérico	Genérico	66,71	77,3/68,9/63,3/58,8

Tabla 5.2: ***BLEUs* obtenidas con el modelo de Catalán *mT5***

Hay que recalcar antes de analizar los resultados que, a pesar de tener bases de datos del dominio institucional para Euskera, en este proyecto no se van a contemplar los resultados, pues el volumen de datos sigue aumentando hoy en día para realizar un entrenamiento específico a partir de los modelos genéricos de Euskera.

A priori deducimos que los resultados no son tan buenos como los del traductor Catalán-Español. El principal motivo, como se comentó previamente, es el tamaño de datos de entrenamiento. El Catalán tuvo el triple de pares de frases de entrenamiento, por lo que comprobamos que es un factor crucial a la hora de obtener un buen modelo. También, la estructura lingüística del Catalán es muy similar al Español, tenemos dualidad entre muchísimas palabras sobre estas lenguas. Sin embargo, el Euskera es una lengua que no tiene ninguna similitud con el Español, por lo que los resultados dificilmente serán mejores.

5.1. Modelos mT5 Euskera

Los modelos que se han entrenado con la lengua Euskera van a presentar una serie de repeticiones a la hora de traducir palabras, como veremos más adelante, factor que empeora la métrica final obtenida. Además, la riqueza lingüística del Español es mucho más amplia que el Euskera, es decir, una palabra en Euskera puede tener varios sinónimos en Español, frente a una única traducción al Euskera. Este factor también va a influir a la hora de obtener la métrica, pues, como se explica en el Anexo A, si las palabras traducida no es idéntica con la original, el resultado empeora. Observamos también que para informativos los resultados son mejores, en el caso del Euskera-Español. Esto se debe a que, en el entrenamiento genérico, algunos de los archivos utilizados tenían algo de contenido de este dominio, por lo que es algo más preciso en este ámbito.

En el caso del Español-Euskera ocurre lo contrario, los resultados son peores. Como veremos en ejemplos de traducción, éste modelo añade varios bucles de repetición a la

hora de traducir, incluso más que su modelo inverso.

Por lo tanto, fue necesario realizar un estudio de los modelos de Euskera para mejorar su calidad de traducción. Se investigó sobre los parámetros ***No repeat ngram size*** y ***Num beams*** y ***Beam width***, los cuales se detallan en el capítulo anterior. Tras realizar varias pruebas con el fichero de ***dev***, se encontraron los valores óptimos de estas variables para mejorar las traducciones, que podemos observar resumidamente en la tabla 5.3.

Source language	*Target language*	*Modelo*	*Dev set*	*Num beams*	*Beam width*	*No repeat ngram*	*BLEU (4-gram)*
Euskera	Español	mT5 Genérico	Genérico	1	5	0	17,23
Euskera	Español	mT5 Genérico	Genérico	3	5	2	18,31
Euskera	Español	mT5 Genérico	Genérico	3	10	2	18,66
Euskera	Español	mT5 Genérico	Genérico	3	10	4	20,87

Tabla 5.3: ***BLEUs* obtenidas en función de los hiperparámetros en *mT5***

La elección de los hiperparámetros adecuados ha conseguido una mejora de 3 puntos de BLEU. El modelo, al traducir, repite secuencias difíciles de eliminar, porque muchas de las repeticiones son sinónimos de las palabras que se vuelven a repetir. Esto no es fácil de eliminar, ya que no es puntual, pero con el parámetro **No repeat ngram size** con valor 4, algunas de las repeticiones se eliminan, en este caso, de conjuntos de 4 palabras.

Los resultados también mejoran cuando aumentamos la anchura de haz ó ***Beam width***, ya que hay más palabras candidatas a ser la traducción a la hora de decodificar. Lo mismo ocurre con ***Num beams***, ya que evitamos que la secuencia correcta se enmascare al elegir palabras con mayor probabilidad en pasos previos.

Otros parámetros a tener en cuenta a la hora de traducir, se trata de los empleados en el ***Pre-processor*** y en el ***Post-processor***, siendo el más importante el máximo tamaño de frase que admitimos que entre a la red, y, si se supera, se trunca lógicamente en trozos, como se explicó en la sección de Análisis. En este proyecto, el tamaño máximo de frase se va a fijar a 200 palabras, ya que las redes han sido entrenadas para secuencias de menos de 400, y la gran mayoría están por debajo de las 200.

Está claro que los resultados numéricos nos dicen que las traducciones no serán demasiado precisas, pero, para comprobarlo, qué mejor que juzgar por nosotros mismos bajo varios ejemplos. A continuación se van a mostrar algunos ejemplos de traducción del Euskera al Español, los cuales se comparan con las traducciones humanas:

FRASE 1: *Hona hemen adibide historiko ospetsu bat: a) gizon bat lehen planoan, espaloian dagoen emakume bat bortizki bultzatzen b) urrutitik ikusita, konturatuko gara gizona emakumea bultzatzen ari dela leiho batetik erortzen ari den objektu pisutsuak zanpa dezakeelako.*

FRASE 1 TRADUCIDA CON MODELO: *Viendo desde lejos un ejemplo histórico famoso: a un hombre en primer plano, un hombre empujar violentamente a una mujer en la acera en un primer plano, una mujer en una acera a empujar de forma violenta b visto a lo lejos, veremos que el hombre empuja a la mujer a través de la acera, nos damos cuenta de que el hombre está empujar a mujer porque puede aplastar el pesado objeto que cae por una ventana.*

FRASE 1 TRADUCIDA POR HUMANO: *Como en este famoso ejemplo histórico: a un primer plano de un hombre que empuja de forma agresiva a una mujer en la acera b visto desde una distancia, descubrimos que el hombre está empujando a la mujer porque un objeto pesado que está cayendo desde una ventana puede aplastarla.*

FRASE 2: *Laburpena Bilbo, lehenik eta behin, beti izan da merkataritza jarduera leku guztiz garrantzitsua Bizkaiko Golkoaren ingurunean, eta era berean, finantzagune nabarmen gisa egituratu zen Ipar Atlantikoko ardatzean.*

FRASE 2 TRADUCIDA CON MODELO: *Bilbao, en primer lugar, siempre ha sido un lugar de actividad comercial muy importante en el entorno del Golfo de Bizkaia, y a su vez se estructura como centro financiero destacado en el eje del Atlántico Norte.*

FRASE 2 TRADUCIDA POR HUMANO: *Bilbao, siempre ha sido, primero, un punto de trasiego comercial de vital importancia en el ámbito cantábrico, al mismo tiempo que se configuró en una relevante plaza financiera dentro del eje noratlántico.*

Como observamos, se aprecian las repeticiones a la hora de traducir, sobre todo en la frase primera. Comprobamos que vuelve a repetirse la secuencia con otras palabras, debido a la falta de datos y que el modelo no es lo suficientemente potente.

En cambio, la segunda frase se comprende perfectamente y la traducción se acerca a la real. Se encuentran más ejemplos de traducciones con los distintos modelos de ***mT5*** en el Anexo B.

5.2. Modelos Helsinki Inglés

Ahora, pasamos a analizar los modelos entrenados de ***Helsinki-NLP***. Al ser modelos pre-entrenados, los resultados van a ser mejores. Las métricas ***BLEU*** obtenidas para cada uno de los modelos entrenados están contenidas en la tabla 5.4.

Como podemos ver, el modelo consiguió mejorar casi 2 puntos de ***BLEU***. La mejora no fue significativa porque, al estar pre-entrenado, el modelo no va a aprender tanto como si lo entrenásemos de cero. En este caso, las traducciones van a ser bastante

Tabla 5.4: ***BLEUs* obtenidas con los modelos de *Helsinki-NLP*** Inglés-Español

Pares de frases de train	0 (modelo base)	10k	20k	40k	100k	200k	400k	800k	1200k	2M	4M	8M	12M	16M	19M
BLEU Score (4-gram)	42.22	41.41	41.04	41.26	41.46	41,83	42,3	42,32	42,41	42,72	42,94	43,16	43,2	43,74	43,31
BLEU Score (unigram)	64.7	65.8	65.6	65.8	65.9	66,2	65,9	66	65,9	66,2	66,2	66.1	66,1	66,4	66,3
BLEU Score (bigram)	46.2	46.8	46.5	46.7	46.9	47,3	47,3	47,4	47,4	47,7	47,9	47,9	48	47,4	48,2
BLEU Score (trigram)	36.4	36.7	36.4	36.6	36.8	37,2	37,4	37,5	37,4	37,8	38	38,1	38,2	38,1	38,4
BLEU Score (4gram)	29.4	29.4	29.1	29.3	29.6	30	30,2	30,3	30,3	30,7	30,9	31	31,1	31,4	31,3

acertadas, si las comparamos con el Euskera.

Al tratarse del Inglés, la cantidad de bases de datos disponibles es enorme. Eso condiciona los resultados de traducción, pues van a ser mucho mejores, ya que el modelo ha sido capaz de entrenar con mucha información. Podemos ver unos ejemplos de traducción a continuación:

FRASE 1: *For example, the EUR 200 and EUR 500 notes, which were not printed in France and in the countries of southern Europe, will reach us, which is in fact already happening with notes of German origin.*

FRASE 1 TRADUCIDA CON MODELO: *Por ejemplo, los billetes de 200 y 500 euros, que no se imprimieron en Francia y en los países del sur de Europa, nos llegarán, lo que de hecho ya está sucediendo con los billetes de origen alemán.*

FRASE 1 TRADUCIDA POR HUMANO: *Por ejemplo, llegarán los billetes de doscientos y de quinientos euros, que ni en Francia ni en los países de del sur de Europa han sido emitidos. Esto ya empieza a observarse con los billetes procedentes de Alemania.*

FRASE 2: *Consequently the funds available to meet its high non-avoidable costs, being mainly financing costs and nuclear liabilities arising from past actions, have been greatly reduced.*

FRASE 2 TRADUCIDA CON MODELO: *En consecuencia, se han reducido considerablemente los fondos disponibles para hacer frente a sus elevados costos inevitables, que son principalmente los costos de financiación y las responsabilidades nucleares derivadas de acciones anteriores.*

FRASE 2 TRADUCIDA POR HUMANO: *Por lo tanto se redujeron considerablemente los fondos disponibles para cubrir sus altos costes fijos, que son principalmente costes financieros y responsabilidades derivadas de actuaciones pasadas.*

Concluimos con que las traducciones son muy precisas, de ahí su alta métrica obtenida. En este caso, al ser un modelo entrenado con multitud de datos, no ocurren repeticiones a la hora de traducir como con el Euskera, por lo que no ha sido necesario modificar hiperparámetros del modelo y de las tareas pre y post traducción. El resto

de ejemplos se contemplan también en el Anexo B.

Capítulo 6

Conclusiones y líneas futuras

Tras la elaboración de este proyecto se han extraído las siguientes conclusiones:

- La traducción automática mediante el uso de ***Deep Learning*** es prometedora y si sigue en desarrollo se obtendrán resultados brillantes.

- La calidad de los datos con los que se entrena la red neuronal artificial condiciona notablemente los resultados que ésta ofrece. Los errores que las secuencias de entrada tienen se ven reflejados en los resúmenes de salida.

- Las estructuras de las lenguas empleadas tanto de origen como de destino influyen mucho a la hora de realizar un modelo, ya que, cuanto menos se parezca lingüísticamente, peores resultados se obtendrán.

- La cantidad de datos disponibles para entrenar la red es muy importante para obtener buenos resultados. Es necesario tener varios millones de pares de frases para entrenar un modelo potente, con la capacidad de decodificar secuencias sin repetir estructuras.

- Los modelos pre-entrenados son una herramienta útil para partir como base a la hora de especializar un modelo, pues, de primeras, se obtienen resultados muy buenos.

- Previo a entrenar la red, es imprescindible dimensionar correctamente el entrenamiento para evitar colapsos de memoria en la tarjeta gráfica de 12 GB de la que se disponía.

Por supuesto, se seguirá trabajando en ello para conseguir optimizarlo al completo y especializar los modelos genéricos en entornos de Instituciones e Informativos.

En cuanto a líneas futuras, destacar que durante la implementación de este TFG, han ido surgiendo algunas ideas con las que experimentar para intentar mejorar el resultado final obtenido. Sin embargo, el tiempo para realizar el proyecto es ajustado y no permite desarrollar estas ampliaciones. A pesar de ello, en este apartado se proponen dichas ampliaciones como líneas futuras para continuar trabajando en este proyecto.

Con los datos disponibles ya hemos conseguido obtener un modelo base de Euskera, pero si estos datos se duplicaran en cantidad, la red tendría mucho más de donde

aprender y obtendría aun mejores resultados. Para ello se propone introducir una técnica denominada ***Data Augmentation*** [24], capaz de obtener ejemplos nuevos a través de los ya existentes. Utiliza técnicas como el cambio de orden del sujeto y el predicado en la oración, de manera que ésta sigue teniendo el mismo sentido, pero es una oración totalmente diferente para la red, por lo que aumentaríamos la cantidad de datos disponibles. Para el caso del Euskera, puede ser una opción muy recomendable.

También, la especialización de los modelos de Euskera en ámbitos Institucionales, donde se enfoca la empresa. Para ello, se siguen obteniendo datos del BOE de Euskadi, recopilando del Parlamento Vasco, sumados a las transcripciones de audio de la empresa, los cuales se han conseguido aumentar considerablemente gracias a la técnica llamada ***Back-Translation*** [25], que se muestra en la figura 6.1.

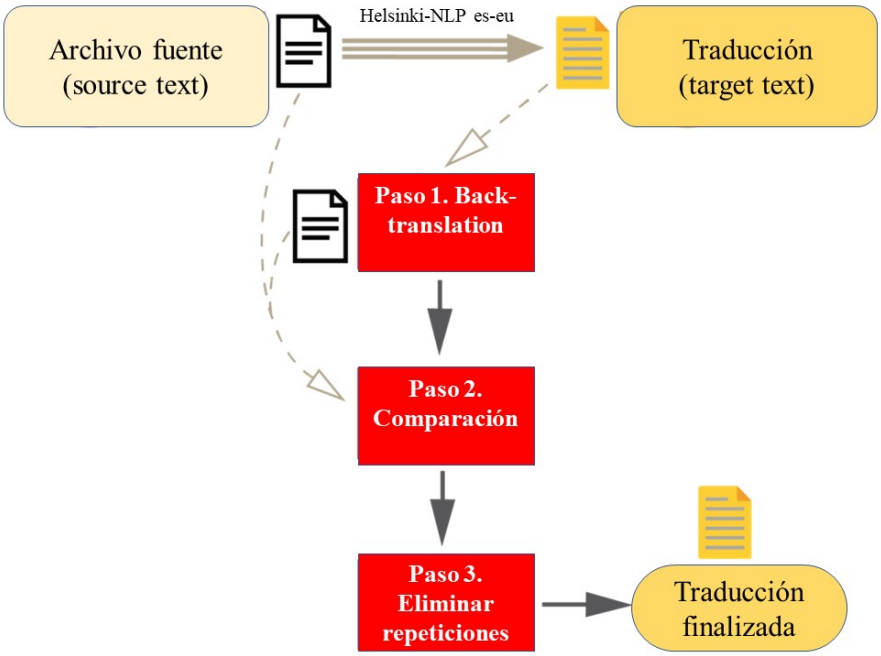

Figura 6.1: Implementación del ***Back-Translation***

Esta técnica consiste en traducir los textos del lenguaje objetivo (Español, en el caso del traductor de Euskera-Español) al lenguaje origen (Euskera) mediante otro modelo pre-entrenado, el cual se ha optado nuevamente por los de ***Helsinki-NLP***, pues las traducciones eran lo suficientemente buenas. Tras esto, tenemos en Euskera el texto original y el ***back-translated***, únicamente debemos limpar las frases que se han traducido igual, ya que estaríamos añadiendo frases repetidas sin utilidad alguna, y ya habremos aumentado el volumen de datos a casi el doble.

También se ha usado esta técnica para traducir plenos transcritos a texto del Parlamento Vasco, de los que se disponía únicamente de los ficheros en Español. De esta forma, seguimos aumentando el volumen de datos para entrenar un modelo específico

con una cantidad considerable de pares de frases.

Recalcar que este TFG ha sido realizado en la empresa ***EtiqMedia*** y la idea es tener un traductor de Euskera lo suficientemente bueno como para traducir transcripciones de plenos del Parlamento Vasco. Para ofrecer el traductor al cliente final, el trabajo va a consistir en optimizar el modelo genérico de Euskera, especializarlo y, tras eso, crear un sistema en producción capaz de introducir las frases transcritas al traductor. Todo ello se dimensionaría de forma que el tiempo de traducción y el consumo de gráfica fueran óptimos.

Todas estas posibilidades podrían servir para diseñar una serie de experimentos, que se realizarían posteriormente al trabajo realizado, para mejorar aún más los resultados.

Capítulo 7

Bibliografía

[1] ¿qué es la traducción automática? traducción automática basada en reglas vs. traducción automática estadística. *Systransoft*, 2022.

[2] M. Corbé. La machine à traduire française aura bientôt trente ans. *Automatisme 5(3): 87-91*, 1960.

[3] J. Hutchins. Traducción automática: pasado, presente y futuro. 1986.

[4] W.N. Locke. *Machine Translation of Languages.* Booth, D.A., eds. (1955). Cambridge, Massachusetts: MITPress. pp. 15-23. ISBN 0-8371-8434-7, 1955.

[5] ALPAC. Lenguaje y mÁquinas: Los ordenadores en la traducciÓn y la lingÜÍstica. 1966.

[6] A. Deoras L. Burget J. Černocký T. Mikolov, S. Kombrink. Rnnlm -recurrent neural network language modeling toolkit. 2011.

[7] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. *Google Research, Google Brain, University of Toronto*, 6 December 2017.

[8] Keyu Duan Dongbo Xi Youngchun Zhu Hengshu Zhu Senio Member Hui Xiong Fuzhen Zhuang, Zhiyuan Qi. A comprehensive survey on transfer learning. *IEEE*, 23 June 2020.

[9] Adam Roberts Katherine Lee Sharan Narang Michael Matena Yanqi Zhou Wei Li Peter J.Lui Colin Raffel, Noam Shazeer. Exploring the limits of transfer learning with a unified text-to-text transformer. *Google, Mountain View, CA, USA*, June 2020.

[10] Adam Roberts Mihir Kale Rami Al-Rfou Aditya Siddhant Aditya Barua Colin Raffel Linting Xue, Noah Constant. mt5: A massively multilingual pre-trained text-to-text transformer. *Google Research*, 11 Mar 2021.

[11] Tomasz Dwojak Hieu Hoang Kenneth Heafield Tom Neckermann Frank Seide Ulrich Germann Alham Fikri Aji Nikolay Bogoychev Andre F. T. Martins Alexandra Birch Marcin Junczys-Dowmunt, Roman Grundkiewicz. Marian: Fast neural machine translation in c++. *Microsoft, Adam Mickiewicz, University in Poznan, University of Edinburgh, Unbabel*, 4 April 2018.

[12] Kyunghyun Cho y Yoshua Bengio Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *CoRR, abs/1409.0473*, 2015.

[13] Tarang Shah. About train, validation and test sets in machine learning. *Towards (Data Science)*, 6 December, 2017.

[14] Ronald J. Williams David E. Rumelhart, Geoffrey E. Hinton. Learning representations by back-propagating errors. *Institute for Cognitive Science, University of California, San Diego, Departament of Computer Science Philadelphia*, 9 October, 1986.

[15] Sebastian Ruder. An overview of gradient descent optimization algorithms. *Insight Centre for Data Analytics, NUI Galway Aylien Ltd., Dublin*, 15 Jun 2017.

[16] Frank La La. ¿cómo aprenden las redes neuronales? *Microsoft Docs, Inteligencia Artificial*, April 2019.

[17] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 2019.

[18] Jason Brownlee. A gentle introduction to early stopping to avoid overtraining neural networks. *Machine Learning Mastery*, 7 December 2018.

[19] Kiprono Elijah Koech. Cross-entropy loss function. *Towards Data Science*, 2 Oct 2020.

[20] Boaz Shmueli. Nlp metrics made simple: The bleu score. *Towards Data Science*, 8 Feb 2021.

[21] Todd Ward Wei-Jing Zhu Kishore Papineni, Salim Roukos. Bleu: a method for automatic evaluation of machine translation. *IBM T. J. Watson Research Center Yorktown Heights, NY 10598, USA*, July 2002.

[22] Ramprasath R. Selvaraju Qing Sun Stefan Lee David Crandall Dhruv Batra Ashwin K Vijayakumar, Michael Cogswell. Diverse beam search: Decoding diverse solutions from neural sequence models. *Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. School of Informatics and Computing Indiana University, Bloomington, IN, USA*, 22 Oct 2018.

[23] Patrick Von Platen. How to generate text: using different decoding methods for language generation with transformers. *Hugging Face Blog*, 18 March 2020.

[24] Jason Wie Sarath Chandar Soroush Vosoughi Teruko Mitamura Eduard Hovy Steven Y.Feng, Varun Gangal. A survey of data augumentation approaches for nlp. *Carnegie Mellon University, Google Research, Mila-Quebec Al Institute, Dartmouth College*, August 2021.

[25] Michael Auli David Grangier Sergey Edunov, Myle Ott. Understanding back-translation at scale. *Facebook AI Research, Menlo Park, CA New York, NY. Google Brain, Mountain View, CA.*, November 2018.

[26]

[27] Jirí Pospicha Daniel Svozil, Vladimír Kvasnicka. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems, Volume 39, Issue 1,*, 1997.

[28] Alumno Apellidos. Citar un tfg. Trabajo fin de grado, Universidad de Zaragoza, 2014.

[29] Alumno Apellidos. Citar un tfm. Trabajo fin de máster, Universidad de Zaragoza, 2014.

[30] Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. 15 February 2014.

[31] Recurrent neural networks. *IBM Cloud Education, IBM Watson Studio*, 14 September 2020.

[32] Jay Alammar. The illustrated transformer. *Github*, 2018.

[33] Hamza Mahmood. The softmax function, simplified. *Softmax Regression, Towards Data science*, 26 November 2018.

[34] Louis Chan. Google's rfa: Approximating softmax attention mechanism in transformers. *Towards Data science*, 27 February 2021.

[35] Justin S. Lee. A math-guided tour of the transformer architecture and preceding literature. *Columbia*, February 2021.

[36] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. *Google AI Language*, 26 May 2019.

[37] Britney Muller. Bert 101 state of the art nlp model explained. *Hugging face blog*, 2 March 2022.

[38] Nick Ryder Melanie Subbiah Jared Kaplan Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry Tom B. Brown, Benjamin Mann. Language models are few-shot learners. 22 July 2020.

[39] Lingyi. Gpt-3, transformers and the wild world of nlp. *Towards Data Science*, 16 Sept 2020.

[40] Tavish Srivastava. A must-read introduction to sequence modelling (with use cases). *Analytics Vidhya,*, 15 April 2018.

[41] Naman Goyal Marjan Ghazvininejad Abdelrahman Mohamed Omer Levy Ves Stoyanov Luke Zettlemoyer Mike Lewis, Yinhan Liu. Bart: Denoising sequence-to-sequence pre-training for natural language generation, traslation and comprehension. 29 October 2019.

[42] Caiming Xiong Richard Socher Romain Paulus. A deep reinforced model for abstractive summarization. *Salesforce Research*, 13 Nov 2017.

Lista de Figuras

Lista de Tablas

Anexos

Anexos A

Métrica BLEU sacrebleu

El procesamiento del lenguaje natural (***NLP***) se ha convertido en un tema candente. Una de las aplicaciones más exitosas es la traducción automática (***MT***), la capacidad de traducir automáticamente texto de un idioma a otro.

Pero nos preguntaremos, ¿cómo vamos a saber si una traducción es buena si no entendemos el idioma destino? O simplemente, ¿cómo se puede evaluar un sistema de traducción automática para saber si sus traducciones son de calidad?

Una forma obvia es usar la evaluación humana [20]. Podemos utilizar el criterio humano para calificar la precisión de la traducción frente a la secuencia original, lo que requeriría expertos que dominen ambos idiomas. Esto sería un método de evaluación muy costoso, además, el tiempo requerido sería muy alto. Por esto, se ha investigado en una forma de evaluar automáticamente el rendimiento de un sistema de traducción automática.

Ahí es donde entra en juego la métrica ***BLEU*** [21], que significa ***Bi-Lingual Evaluation Understudy***. Es una forma popular y económica de medir automáticamente el rendimiento de su modelo de traducción.

El enfoque funciona contando los ***n-gramas*** coincidentes en la traducción del candidato a n-gramas en el texto de referencia, donde 1-gram o ***unigrama*** sería cada token y una comparación ***bigrama*** sería cada par de palabras. La comparación se realiza independientemente del orden de las palabras.

También se modifica el recuento de n-gramas coincidentes para garantizar que tiene en cuenta la aparición de las palabras en el texto de referencia, sin recompensar a una traducción candidata que genere una abundancia de palabras razonables. Esto se conoce en el documento como precisión modificada de n-gram.

La puntuación se usa para comparar oraciones. Una partitura perfecta no es posible en la práctica, ya que una traducción tendría que coincidir exactamente con la referencia. Esto ni siquiera es posible para los traductores humanos. El número y la calidad de las referencias utilizadas para calcular la puntuación BLEU significa que la

comparación de las puntuaciones entre conjuntos de datos puede resultar problemática. Veamos que una oración puede tener muchos significados y se puede traducir de múltiples maneras con otras estructuras y sinónimos, donde la puntuación será baja, pero realmente el sentido semántico es el mismo.

Tras haber introducido esta métrica, veamos cómo realmente evalua la traducción. En este proyecto, al estar escrito puramente en ***Python***, se ha utilizado la librería ***Sacrebleu***, que incluye esta métrica. La forma de uso es llamar a la función de cálculo de la métrica, que tiene como entrada una lista de frases traducidas y otra lista con las traducciones buenas, la lista de ***ground truth***.

Los cálculos de la puntuación ***BLEU*** permiten especificar la ponderación de n-gramas diferentes en el cálculo de la puntuación. Esto le da la flexibilidad para calcular diferentes tipos de puntuación BLEU, como las puntuaciones individuales y acumulativas de n-gramas.

BLEU funciona calculando la precisión, es decir, la fracción de tokens del candidato que aparecen o están çubiertos"por las referencias. El valor de la puntuación ***BLEU*** siempre es un número entre 0 (peor) y 1 (mejor). Ésta métrica no tiene en cuenta el orden de las palabras a la hora de comparar la secuencia original con la traducida, por lo que, a veces, puede ser un problema, pues cambiar palabras de orden pueden alterar el significado de la frase original drásticamente.

Para el cálculo de la ***BLEU***, se utiliza la precisión modificada, la penalización por brevedad y por repeticiones.

La precisión modificada indica la relación entre los n-gramas comunes de la o las frases candidatas y la frase original. La relación es la siguiente:

$$P = \frac{n - gramas_{comunes}}{n - gramas_{candidata}} \tag{A.1}$$

Para cada ngrama único, contamos su frecuencia máxima en cada una de las oraciones de referencia. El mínimo de este recuento especial y el recuento original se denomina recuento recortado. Es decir, el recuento recortado no es mayor que el recuento original. Luego usamos este conteo recortado, en lugar del conteo original, para calcular la precisión modificada. Es una forma de contar los ngramas que son únicos.

Veamos un ejemplo de ***BLEU*** para dos referencias y dos candidatas a traducción, como muestra la tabla A.1. En este caso, la precisión de unigramas vendrá dada por las palabras que se compartan entre las candidatas y de los bigramas entre grupos de dos palabras. También cuentan para el cómputo de la BLEU los trigramas y 4-gramas, aunque no se tengan en cuenta en estos ejemplos.

Para la frase candidata 1 tenemos las ocurrencias resumidas en la tablas de los

Tabla A.1: Ejemplo de precisión de BLEU

Referencia	el juez expuso el veredicto
Candidata 1	el el el el el el el
Candidata 2	el juez el juez el veredicto

unigramas y bigramas A.2.

Tabla A.2: Ocurrencias y resultados Candidata 1 unigramas

Unigrama único	Conteo	Conteo recortado	Precisión unigrama
el	7	2	0,286

Tabla A.3: Ocurrencias y resultados Candidata 1 bigramas

Bigrama único	Conteo	Conteo recortado	Precisión bigrama
el el	0	0	0

La precisión obtenida para los unigramas (A.2 en esta oración candidata va a ser 2/7, ya que dos de las palabras de la traducción coinciden en la frase de ground truth. La de los bigramas (A.3) va a ser nula, pues en la oración de referencia no hay ninguna ocurrencia de la secuencia "el el".

Los resultados de la candidata 2 se va a realizar de la misma manera, analizando los unigramas y bigramas únicamente.

Tabla A.4: Ocurrencias y resultados Candidata 2 Unigramas

Unigrama único	Conteo	Conteo recortado	Precisión unigrama
el	3	2	0.67
juez	2	1	
expuso	0	1	
veredicto	1	0	

Tabla A.5: Ocurrencias y resultados Candidata 2 Bigramas

Unigrama único	Conteo	Conteo recortado	Precisión bigrama
el juez	2	1	1
juez expuso	0	1	
expuso el	0	1	
el dictamen	1	0	

En la tabla A.4 vemos que se obtiene una precisión de unigrama de 4/6, tras sumar las ocurrencias únicas y dividirlas entre el número total de ocurrencias en la secuencia candidata 2.

Sin embargo, la tabla A.5 vemos que se obtiene una precisión de unigrama de 1. Esto, aunque parezca raro, no va a ser así finalmente, ya que van introducirse penalizaciones por repetición y por longitud.

La BLEU final se calcula utilizando varias precisiones modificadas de ngram. Se expone en la ecuación siguiente:

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log P_n) \tag{A.2}$$

se utiliza la media geométrica para los N ngramas que se vayan a utilizar. Cada ngrama tendrá un peso: $w_n w_n$

Donde típicamente $w_n = \frac{1}{N}$.

También se introduce un penalizador por brevedad de las frases candidatas:

$$BP = \begin{cases} 1 \text{ si } c > r \\ e^{1-\frac{r}{c}} \end{cases} \tag{A.3}$$

Donde c es la longitud de la frase candidata y r la longitud de la frase de referencia.

Aquí, la mejor longitud de coincidencia es la longitud de oración de referencia más cercana a las oraciones candidatas. Por ejemplo, si hay tres referencias con longitudes de 12, 14 y 17 palabras y la traducción candidata es una concisa 13 palabras, idealmente, la mejor longitud de coincidencia podría ser 12 o 14, pero elegimos arbitrariamente la más corta, que es 12.

Por lo tanto, tras computar las precisiones de los ngramas para cada una de las frases candidatas en la expresión A.2, obtendríamos la BLEU final. Esto se repetiría para cada una de las candidatas frente las referencias.

Anexos B

Ejemplos de traducción y mediciones

El punto clave de este proyecto es obtener traducciones de calidad mediante técnicas de ***Deep Learning***. En este apartado se van a mostrar las distintas pruebas de traducción realizadas, con ejemplos y mediciones, para poder analizar lo máximo posible los modelos que se han entrenado.

B.1. Modelo mT5 Euskera-Español

Para este modelo en específico, se entrenaron alrededor de 20 épocas y se consideró finalizado mediante la técnica de ***Early-Stopping***. En el capítulo de Análisis de resultados se visualizaron un par de traducciones de este modelo, por lo que vamos a seguir desde ese punto. Se van a analizar frases de distintos tamaños, sobre todo, de transcripciones de audio pertenecientes la televisión vasca ETB, que fueron transcritos por la empresa. Éstos datos sólo están disponibles en Euskera, no tienen su traducción al Español, por lo tanto las traducciones se compararán con el traductor de Google, para ver cuánto de lejos está nuestro modelo. En estas transcripciones hay errores, los cuales se verán reflejados en las traducciones, como frases en Español, cuando deberían estar en Euskera.

FRASE 1.1: *Anna Gabriel haurraren ustezko hiltzailea baderatzi an epaituko du; zazpi emakume eta bi gizonezkok osatutako herri epaimahai batek zortzi urte zituen eta haren bikotekidearen semea zen eta behin eta berriro kolpatu zuen Iriarte.*

FRASE 1.1 TRADUCIDA CON MODELO: *Un jurado popular formado por siete mujeres y dos hombres, Anna Gabriel Anna Gabriel, supuestamente asesina del niño Anna Gabriel, juzgará si el presunto asesino del niño es de nueve; un jurado popular de siete hombres y ocho años de edad, y formado por un jurado de pueblo formado por una pareja de ocho años y que era hijo de su pareja y que golpeó una y otra vez Iriarte.*

FRASE 1.1 TRADUCIDA POR GOOGLE: *Anna Gabriel será juzgada en relación con el presunto asesinato del niño; un jurado del pueblo de siete mujeres y dos hombres tenía ocho años y era hijo de su pareja y golpeó repetidamente a Iriarte.*

FRASE 1.2: *Albistegien denboraldi berria zuen konfiantzari hutsik ez egiteko ahalegina hitzematen dizuegu aurten ere eta hutsik egin gabe euskal haur eta gazteek hasi dute gaur ikasturtea Euskal Autonomia Erkidegoan hori bai matrikulazio azken hamabost urteetan lehendabiziko behera egin dute Amaia Ibáñez matrikulazioak ehunekoa 92 eta hogeita 13 jaitsi dira aurten jaiotze-tasa baxua dela-eta eta Haur Hezkuntzan bakarrik ez Lehen Hezkuntzan ere hasi da ikertzen beherakada hori.*

FRASE 1.2 TRADUCIDA CON MODELO: *Hoy os damos a ustedes un esfuerzo por no hacer vacío a vuestra confianza la nueva temporada de noticias Los niños y jóvenes vascos han comenzado este curso en la Comunidad Autónoma Vasca y sin faltar os vamos a anunciar el esfuerzo de no hacer ningún vacío al esfuerzo que tenéis que hacer a la confianza, y sin ningún error los niños, niñas y juveniles vascos han empezado hoy el curso en Euskal Herria, así como las matriculaciones en los últimos quince años han bajado la primera matriculación de Amaia Ibáñez, el 92 por ciento y los treinta y 13 por descenso de la tasa de natalidad por bajada, y este año se ha empezado a investigar solo en Educación Infantil y también en Educación Primaria.*

FRASE 1.2 TRADUCIDA POR GOOGLE: *Os recordamos que los niños y jóvenes vascos han comenzado hoy el curso escolar en la Comunidad Autónoma Vasca, siendo este el primer descenso de la matrícula Debido a la baja tasa, este descenso ha comenzado a estudiarse no solo en Educación Infantil sino también en Educación Primaria.*

FRASE 1.3: *Edozein kasutan gaur ia 94 eta hirurofeita hamabost mila ikaslek hasi dute ikasturtea Euskadin eta aukeratzen duten eredua 96 ereduaren alde handiarekin gainera bere osotasunean hartuta iaz baino ikasle gutxiago izango ditu. Globalean aldaketa oso txikia da baina inflexio-puntu bat erakusten digu matrikulazioaren dagokionez euskaltegietan ere eta kontrolatzeko Euskal euskaltelebista. Albistegien denboraldi berria ere gaur bertan jarri dugu abian ikusten duzuenez platoan aldaketak egin ditugu badira eta izango dira. Baliabide tekniko biak ere goi goi mailakoak baina beti bezala gure helburu nagusia zuen konfiantzari eustea da. Betiko legez hurbileko eta egiazko informazioa eskaini. Eta. Hala Harvey gaurko Kontxako garaipenaren ostean biharamun osoa izango du bai baita ere biek ala biek balantzea egin zuten*

atzo Kontxako irabazleek banderak ederto astindu zituzten. Atzo eta lortutako balentria horren Saria eskuratzen ari dira oraindik eta hala jarraituko dute Adegi Gipuzkoako enpresen elkarteak.

FRASE 1.3 TRADUCIDA CON MODELO: *En cualquier caso, hoy casi 94 y tresrofeita quince mil estudiantes han iniciado el curso en Euskadi en Euskadi, y el modelo que elijan en Euskadi es muy pequeño, pero con un gran partido del modelo 96 y controlar Euskal euskaltelebista. También hemos puesto en marcha hoy mismo la nueva temporada de informativos que hemos hecho cambios en el plató en cuanto a la matriculación, y con un punto de inflexión, aunque como siempre nuestro principal objetivo es mantener nuestra confianza en los euskaltegis, y como siempre, el modelo de matriculación es un cambio en el platón, y además de tener en su totalidad el modelo 96. En general, el cambio es muy bajo, pero como siempre nos muestra un punto inflexivo en el plato, y que el modelo elegido es un poco más alto, pero también como siempre el objetivo principal es mantener la confianza que tenía. En este sentido, ambos recursos técnicos han hecho y van a tener en cuenta que el modelo que hemos elegido en Euskadi y con el modelo 96 en su conjunto. El nuevo periodo de informativos hemos presentado hoy mismo hemos tenido un cambio de plató, pero con una gran diferencia del modelo 96 como siempre en el euskera. También en el mundo de los informativos, hemos visto que el modelo hemos evolucionado y serán menos estudiantes en Euskadi. La nueva jornada de informativos son muy pequeños, pero nuestro objetivo siempre es mantener su confianza en el plato y también*

FRASE 1.3 TRADUCIDA POR GOOGLE: En cualquier caso, hoy han comenzado el curso escolar en Euskadi casi 94 y setenta y cinco mil alumnos y el modelo que elijan tendrá menos alumnos que el año pasado, con una gran diferencia con el modelo 96. En conjunto, el cambio es muy pequeño, pero marca un punto de inflexión en cuanto a la matrícula en las escuelas vascas y en el control de la Euskal euskaltelebista. También hemos lanzado una nueva temporada de noticias hoy, como puede ver si hemos realizado algún cambio en el conjunto. Ambos recursos técnicos son de primera pero como siempre nuestro principal objetivo es mantener la confianza que tenías. Proporcionar información siempre actualizada y veraz. Y. Sin embargo, Harvey tendrá un día completo tras la victoria de hoy en La Concha. La asociación empresarial Adegi Gipuzkoa sigue recibiendo ayer el galardón por esta hazaña y lo seguirá haciendo.

FRASE 1.4: *Covid pandemiaren txarrena pasa ondoren hala erakusten du Ikuspegi Inmigrazioaren Behatokiak egindako azken txostenak ehunekoa hirurofeita sei an dago tolerantzia indizea duela bi urte baino 6, gorago immigrazioa ez dugu orain arazo gisa*

ikusten ez baitugu covid krisiaren jatorria quien lotzen Arkaitz Fullaondo ikuspegiko ikertzailea euskal gizartearentzako immigrazioak ez da arazoa aurreko krisian arazo bat izan zen da bazegoen euskal gizartearen sektore bat migrazio eta grisaren arteko erlazio eza ezartzen zuena baina krisi honetan ez Euskadi Irratian gaur goizean jaso dugun beste albistea hauxe ere bai Bergarako Udalak baliogabetu egin duela papergintza hondakinak tratatzeko lantegi proiektuari uztailean emandako hirigintza baimena Pablo Gring enpresak Udalari aurkeztutako jarduera proiektua eta Eusko Jaurlaritzari aurkeztutakoa ez baitator ez baitator bat Euskadi Irratiko Faktorian. Halaxe iragarri digu Gorka Artola. Bergarako alkateak balioaz jabetu egin duela ikusita ez dagoela argi zein den aurrera eramaten den aktibitate hori enpresak berriz egin beharko da eskaera Udalean eta bertan benetan garatu nahi da jarduera hori zehaztu ikusiko da Bergarako hirigintza arautua hitzarekin bat gatozen edo ez eta eskari gehiago proiektuaren ingurumen baimena tramitatzen ari den Jaurlaritzako sailari alegazioak aurkezteko epea luzatzeko eskatu dio Udalak eta orain arte Confidencial diren hainbat txosten publiko egiteko eta Asturiasen Javier ardi Inesen hilketaren ustezko bi eragile bizkaitarrek goitik behera ukatu dituzte epailearen aurrean akusazio guztiak Pedro nieva ustez hilketa antolatu zuena ukatu egin du ardi no es ético inolako gorrotorik zuenik.

FRASE 1.4 TRADUCIDA CON MODELO: *El último informe elaborado por el Observatorio de Inmigración señala Ikuspegi Inmigración ha demostrado que el índice de tolerancia tresrofeita seis por ciento se encuentra tres veces 6 por ciento es un problema para la sociedad vasca no vemos como problema en la crisis anterior no vemos el origen de la crisis coronavirus quien vincula a la inmigración es un problema porque existía un sector vasco que establecía la falta de relación entre la migración y el gris, pero no en esta crisis no en Euskadi Irratia hemos recibido esta mañana otra noticia que el Ayuntamiento de Bergara ha anulado esta misma mañana la autorización urbanística concedida al proyecto de taller de tratamiento de residuos de papelería presentado por la empresa Pablo Gring al Ayuntamiento en julio y presentado al Gobierno Vasco en julio el proyecto de actuación del Ayuntamiento a Bergara y el proyecto presentado al Ayuntamiento al Gobierno de Bergara al Proyecto de Taller de Tratamiento de Residuos Industriales de papelería que ha presentado el Ayuntamiento Pablo Ging y el Gobierno vasco que no viene a la presentación del Proyecto de Obras de Trabajo para tratar residuos papelería a la empresa Bergara, y el Ayuntamiento en Bergara ha cancelado la presentación de proyecto de papelería al Ayuntamiento y el de Gobierno a la ciudadanía de Bergara a julio el permiso de urbanización presentado al ayuntamiento y que presentado al Ejecutivo vasco en julio, y no viene en la factura de papelería el Ayuntamiento ha anulado el proyecto Así nos lo ha anunciado Gorka Artola, Gorka Antola. El Ayuntamiento ha pedido al Ayuntamiento que, a la vista*

de que el alcalde de Bergara ha tomado conciencia del valor de la actividad que se lleva a cabo en el Ayuntamiento y en el mismo se ha solicitado que la empresa vuelva a realizar dicha actividad y se va a desarrollar de forma clara a la hora de realizar informes públicos que se ajusten a la palabra urbanismo reglado de Bergara o no y más demandas al departamento del Gobierno que está tramitando la autorización ambiental del proyecto, y que se vaya a concretar la actividad de la empresa para realizar o no la palabra Urbanismo regulado para Bergara y se quiere ampliar el plazo de presentación de alegaciones al Departamento de Gobierno que esté tramitando el proyecto de autorización ambiental, y en Asturias, los presuntos agentes vizcaínos del asesinato de Javier oveja Inés en Bergara ha solicitado a la Alcaldesa de Bergara, el Ayuntamiento ha solicitado la recepción de la solicitud para que la empresa sea consciente del valor que se ha llevado adelante con la palabra Bilbao urbanismo regulado o no, y se trata de realizar una nueva solicitud para realizar la actividad en el ayuntamiento para que se pueda llevar adelante la actividad a la que se encuentra de acuerdo con el título de Bergara urbanismo reglada y más reclamaciones al Departamento del Gobierno en tramitación del proyecto para realizar una serie de Informes Públicos Confidenciales, y además se quiere desarrollar en dicho Ayuntamiento para que

FRASE 1.4 TRADUCIDA POR GOOGLE: El último informe del Observatorio de la Inmigración sobre el Covid muestra que el porcentaje del índice de tolerancia es seis veces superior al de hace dos años El problema era que en la crisis anterior había un sector de la sociedad vasca que establecía una falta de relación entre la migración y el gris. , pero en esta crisis, no la otra noticia que recibimos esta mañana en Euskadi Irratia de que el Ayuntamiento de Bergara El proyecto de actividad presentado por Pablo Gring al Ayuntamiento y el presentado al Gobierno Vasco no está de acuerdo con Euskadi Irratiko Faktoria. Eso nos dijo Gorka Artola. Dado que el alcalde de Bergara se ha dado cuenta del valor, no está claro cuál será la actividad.El Ayuntamiento ha pedido al Gobierno que amplíe el plazo para presentar alegaciones y que haga públicos varios informes hasta el momento Confidencial y los dos presuntos agentes vizcaínos. del asesinato de Javier Inés en Asturias han negado por completo todos los cargos ante el tribunal que no tenía odio ético.

FRASE 1.5: *Bai Elkarrekin Podemos-IU izan zen. Gainera aurtengo aurrekontuak adostu etik hurbilen egon zen. Koalizioa orain ere prest dago negoziatzeko ez dute oraindik tematuta zenbat suposatuko duen proposatzen dituzten neurriak aurrera eramatea. Baina beharrezkotzat jotzen dute Osakidetza indartzea pandemia garaian kontratatu diren 4.000 langileri lanean jarraitzeko aukera emanez edota*

zerbitzuak indartzea hezkuntza gehiago inbertitzea. Sare publikoko eskoletan energia berriztagarrien aldeko enpresa publikoa sortzea eta koalizioaren esanetan dauden murrizketa sozialak aurrekontu hauek ez dira nahikoak ditugun erronkei aurre egiteko ez zerbitzu publikoak indartzeko ezta ere ekonomia sustatzeko eta buelta bat eman behar zaiola dena dela gu prest negoziatzeko gure proposamenak egingarriak direlako beharrezkoak direlako eta horrekin gainera gastu sozialean egin diren murrizketei buelta emateko Elkarrekin Podemosi aurkeztuko duen proposamena. Beraz azken eratzen ari dira orain erreforma fiskala ere agertuko da eta Nafarroan EH Bildurekin akordioa edo aurrekontu akordioa bideratuta duela ekingo dio bihar María Chivite. Lehendakariak foru erkidegoaren egoerari buruzko eztabaidari Chiviteren hitzaldiaren ardatz ditu Javier Remírez bozeramaileak pandemiak eragin duen krisiak krisi ekonomiko eta sozialak gainditzeko eskua luzatuko die lehendakariak Foru Parlamentuko talde guztiei erasoa Iruñea arratsalde on arratsalde on eztabaida politikoa antzuak albo batera utzita proposamenak eginen ditu Maria Chivite foru gobernuburuak osoko bilkura berezian hezkuntzan osasunean gizarte-politiketan lurralde kohesioan aurrerapausoak emateko proposamenak. Eta bide hori elkarrekin egiteko gonbitea egin die foru Parlamentuko talde guztiei hori delakoan Gobernuaren lehentasuna ez ezik herritarren gehiengoaren aurreratu du Javier Remírez. Presidenteordeak es un efecto positivo en el enfoque mano tendida y creo que sinceramente lo que motiva hori bai Remirezek ez du ezkutatu aurrekontuen eztabaidaren harira Gobernuarengandik gertuago ikusten duela EH Bildu Navarra Suma baina hori berretsi du Enrique Maya eskuineko koalizioko presidenteorde den Iruñeko alkateak Radio Euskadin Bildu el Gobierno Chivite maiaren hitzetan Chivite lehendakaria den arren EH Bilduk agintzen du Nafarroan giro horretan eginen da.

FRASE 1.5 TRADUCIDA CON MODELO: *Fue Bai Elkarrekin Podemos-IU. Además, además, estuvo más cerca del acuerdo de los presupuestos de este año que se acordaron los Presupuestos para este año. La coalición de coalición aún está dispuesta a negociar para negociar, aún no están aún dispuestas a llevar a cabo medidas que propongan cuánto supondrán las que proponen. Pero consideran necesario reforzar Osakidetza dando la oportunidad de seguir trabajando a las 4.000 trabajadoras y trabajadores contratados durante la pandemia y/o reforzando los servicios reforzando la inversión de más educación en la inversión en educación. Crear una empresa pública a favor de las energías renovables en las escuelas de la Red Pública en redes públicas creando una empresa pública en favor de las renovables y las reducciones sociales que según la coalición están señaladas por la Coalición, estos presupuestos no son suficientes para hacer frente a los retos a los que nos enfrentamos, ni para reforzar los servicios públicos ni para fomentar la economía y dar una vuelta a nuestra economía,*

porque es necesario que nuestras propuestas sean factibles para negociar con nosotros, y con ello, además, compartir con Elkarrekin Podemos la propuesta que presentará a Podemos para dar un giro a los recortes que se han producido en el gasto social. Por lo tanto, ahora están formando últimamente, ahora aparecerá también la reforma fiscal y en Navarra también aparecerá la reforma fiscal, y mañana María Chivite comienza mañana con EH Bildu el acuerdo o acuerdo presupuestario en Navarra dirigido a EH Bildu. El portavoz Javier Remírez centra el discurso de Chivite al debate sobre la situación de la comunidad foral sobre la crisis económica y social La crisis que ha provocado la pandemia en el debate de la Comunidad Foral El portavoz, Javier Remérez, centra la intervención del portavoz del portavoz Chivite en la conferencia de Chivite sobre el debate sobre el estado de la sociedad foral El Lehendakari extenderá la mano a todos los grupos del Parlamento Foral para superar las crisis económicos y sociales debido a la pandemia La crisis provocada por la pandemia El Lehendakari ofrecerá a todas las entidades del Parlamento foral la mano de superar la crisis de la crisis que la pandemia ha provocado en la Comunidad foral La presidenta el Lehendakari alargará la mano para superar crisis sociales y económicas La crisis en la que la crisis se ha producido en la pandemia el ataque a los grupos parlamentarios forales La presidenta del Lehendakari dará mano a los diferentes grupos del parlamento foral a los ataques a Pamplona buenas tarde buenas tardes buena tarde buena noche buena buena debate político dejando a un lado los estériles debate político, el Lehendakari propondrá a todos y todas las formaciones del Parlamento a favor del ataque Iruñea anoche anoche la buena tarde la buena anoche el debate político estéril el Presidente del Gobierno foral María Chivite propondrá propuestas para dar pasos adelante en la cohesión territorial de las políticas sociales en El Parlamento Foral ha invitado a todos los grupos del Parlamento foral a realizar juntos ese camino y ha invitado, además de la prioridad del Gobierno, a todas las y los grupos parlamentarios forales a hacer juntos este camino, y ha hecho una invitación a todos y todas y todos los equipos de la Diputación Foral a hacer este camino juntos, en este sentido, ha invitado no sólo a la prioridad de Gobierno, sino a la mayoría de la ciudadanía, ha adelantado Javier Remírez, Javier Remérez. El vicepresidente vicepresidente es un efecto positivo en el enfoque mano tendenciida y creo que sinceramente lo que lo que motiva eso sí Remirez no ha ocultado que ve más cerca del Gobierno al hilo del debate presupuestario EH Bildu Navarra Suma pero lo ha ratificado el alcalde de Pamplona, vicepresidente de la coalición derecha, Enrique Maya, vicepresidente del Gobierno de Pamplona en Radio Euskadi Bildu el Gobierno Chivite en palabras del vicepresidente de Pamplona Iruñea, Bildu el gobierno Chivite maya, que aunque Chivite es presidente, EH Bildu mantiene EH Bildu gobi Chivite es Presidente de Chivite en Nafarroa, aunque EH Bildu ordena

que se haga en ese ambiente en Nafarroa.

FRASE 1.5 TRADUCIDA POR GOOGLE: *Sí, fue Juntos Podemos-IU. También estuvo cerca de acordar el presupuesto de este año. La Coalición ahora está lista para negociar, pero aún no han determinado hasta dónde llegará. No obstante, consideran necesario reforzar Osakidetza permitiendo que los 4.000 trabajadores contratados durante la pandemia sigan trabajando, o reforzar los servicios invirtiendo más en educación. La creación de una empresa pública de energías renovables en los colegios públicos y los recortes sociales que la coalición dice no son suficientes para afrontar los retos a los que nos enfrentamos ni para reforzar los servicios públicos ni dinamizar la economía.Además, presentará una propuesta a Elkarrekin Podemosi para revertir las reducciones en el gasto social. Así que se están ultimando, ahora también aparecerá la reforma fiscal y María Chivite empezará mañana con un convenio o acuerdo presupuestario con EH Bildu en Navarra. Javier Remírez, vocero de la crisis de la pandemia, extenderá la mano para superar la crisis económica y social las propuestas para avanzar en la cohesión territorial en las políticas sociales en educación, salud, salud en una reunión especial. Y Javier Remírez ha llamado a todos los grupos del parlamento provincial a emprender juntos este camino, no solo porque es una prioridad para el Gobierno, sino también para la mayoría de la ciudadanía. La Vicepresidenta incide positivamente en el planteamiento extendido y creo que sinceramente motiva a ambos Remírez no ha ocultado que ve a EH Bildu Navarra Suma más cerca del gobierno en el debate presupuestario Según el gobierno maya de Chivite, aunque Chivite es el presidente, EH Bildu promete que tendrá lugar en Navarra.*

Observamos en las frases traducidas por nuestro modelo aún están lejos de las traducciones de Google, si las comparamos. Nuestro modelo, de momento, añade repeticiones a la hora de traducir, repitiendo las secuencias en varias ocasiones. Como se ha comentado, esto se debe a la inconsistencia del modelo debido a la escasez de datos de entrenamiento, a pesar de utilizar hiperparámetros óptimos y un postprocesado que intente eliminar estas repeticiones. El tamaño máximo de palabras de la frase de entrada se ha fijado a 200, de forma que, si se supera ese tamaño, el preprocesado trunca la frase con lógica.

Debemos tener en cuenta también que se trata de un modelo genérico que sirve como base, no va a ser un modelo definitivo para entornos específicos.

Los tiempos y consumos se recogen en la tabla B.1 y gráficamente en la imagen. Podemos deducir que, una vez se supera el tamaño máximo fijado a 200 palabras, el modelo trunca las frases y los resultados son mucho más prometedores en cuanto a tiempo y consumo. Estas mediciones se realizan en media, es decir, se realiza un promediado del tiempo de traducción de cada frase. Si la secuencia no supera el tamaño

máximo, se introduce a la red, separando por saltos de línea. Y, si por el contrario, se supera la longitud, la frase se trunca. Por eso, las frases que no superan 200 palabras el tiempo y consumo es mucho más alto, pero si se supera éste valor y se truncan, los resultados son mejores, ya que estamos introduciendo frases truncadas más pequeñas.

Realmente, el tiempo total va a ser el mismo, pero para saber lo que consume y tarda cada secuencia, se realiza una media en tiempo y consumo por frase truncada.

Tabla B.1: Tiempos y consumos de las frases analizadas, con extit**mT5** Euskera-Español

Frase	Tiempo de ejecución medio (s)	Consumo medio (MB)	Número de palabras
Frase 1	2,99	348	33
Frase 2	5,33	468	65
Frase 3	8,98	590	138
Frase 4	5,86	338	228
Frase 5	3,74	196	295

B.2. Modelo mT5 Español-Euskera

Con el modelo inverso al anterior, se realizó el mismo entrenamiento, con la diferencia que ahora la lengua fuente es el Español y la objetivo el Euskera. La forma de analizar las traducciones de este modelo se va a hacer de la siguiente manera:

Partiremos de secuencias en Español, las cuales traduciremos al Euskera y volveremos a traducir al Español, para ver qué tal vuelven a su modelo origen. Ésta prueba se va a realizar con ambos modelos de ***mT5***.

Se van a utilizar datos del Parlamento Vasco transcritos. No se va a contemplar, en este caso, los consumos y tiempos empleados.

FRASE 2.1: *Las distintas autoridades de supervisión, designadas por las administraciones competentes en sus respectivos ámbitos territoriales, y la Oficina, que consolidará toda la información, pondrán a disposición del público el resultado del proceso de supervisión, que se publicará en las correspondientes plataformas de contratación.*

FRASE 2.1 TRADUCIDA - EUSKERA - ESPAÑOL:

Las diferentes autoridades de Estudio de Estudio, las distintas autoridades de inspección designadas por las administraciones competentes en sus respectivos

ámbitos territoriales, las autoridades de supervisión designadas por la Administración competente en sus respectivas áreas territoriales, y la Oficina que fije toda la información, la que refuerce toda la información y la oficina que consolide toda la información pondrán a disposición del público el resultado del proceso de inspección, el resultado de la inspección que se publicará en las correspondientes plataformas de contratación y el resultado de los procesos de inspección publicados en las plataformas correspondientes de contratación, que se publicarán a la vista del público, que será publicado en las respectivas plataformas contractuales, que se publicará en las Plataformas de Contratación correspondientes, que se hará pública en las mismas plataformas que correspondan, y que será la publicación de la plataforma de contratación correspondiente.

FRASE 2.2: *A estos efectos, la Intervención General de la Administración del Estado, y los órganos equivalentes a nivel autonómico y local, remitirán anualmente a la Oficina Independiente de Regulación y Supervisión un informe global, que se hará público dentro del mes siguiente a su recepción, con los resultados más significativos de su actividad de control en la contratación pública.*

FRASE 2.2 TRADUCIDA - EUSKERA - ESPAÑOL:

A estos efectos, la intervención general de la Administración del Estado y la Intervención General de la Administración del Estado, así como las Intervenciones Generales de la Administracion Estatal y los órganos equivalentes de autonomía y ámbito local equivalentes a órganos equivalentes autonómicos y locales, y las entidades equivalentes del ámbito autonómico y local, y el órgano equivalente a la Administración estatal, y la Administración General del Estado, intervención general y administración estatal equivalente, y entes equivalentes, a nivel autonómico-local y autonómico-locales, remitirán anualmente un informe global a la Oficina Independiente de Regulación y Inspección, y un informe general anual a la oficina independiente de regulación y inspección, que se publicará anualmente al mes siguiente a su recepción, con los resultados más significativos de su actividad de control en la contratación pública, a los efectos de dichos efectos a los que se refiere, a la Secretaría Independiente de Régimen e Inspección, que a su vez será publicado al mes siguiente al de la recepción de dicho informe, con las conclusiones más relevantes de su actuación de control en contratación pública.

FRASE 2.3: *La ampliación del ámbito, se aprobará mediante Orden del Ministro de Hacienda y Función Pública, a propuesta de la Oficina Independiente de Regulación y Supervisión de la Contratación, previo informe de la Comisión Delegada del Gobierno para Asuntos Económicos.*

FRASE 2.3 TRADUCIDA - EUSKERA - ESPAÑOL:

La ampliación del ámbito de ampliación se aprobará mediante Orden del Ministro de Hacienda, por Orden de Ministro de hacienda y función pública, por Orden del ministro de Hacienda y Función Pública, a propuesta de la Oficina Independiente de Regulación y Supervisión del Área de ampliación del Área Ampliación del Área, mediante Orden de la Ministra de Hacienda en Orden de la Hacienda, y de la Función pública, propuesta por la Oficina independiente de Norma y Estudio de la Contratación, mediante Propuesta de Oficina de Norma e Inspección Independiente de Contratación y de Estudio Independiente, tras la elaboración del informe de la Junta de Delegación del Gobierno para Asuntos Económicos, tras el informe de la Comisión Delegada del Gobierno de Gestión Económica, del Comité Delegado del Gobierno en materia de problemas económicos, después del informe del Consejo Delegado de Gobierno de asuntos económicos del Gobierno sobre Problemas Económicos.

FRASE 2.4: *Si la Administración o la entidad destinataria del informe se apartara de las recomendaciones contenidas en un informe preceptivo de la Oficina, deberá motivarlo en un informe que se incorporará al expediente del correspondiente contrato y que será objeto de publicación en su perfil de contratante y en la Plataforma de Contratación del Sector Público.*

FRASE 2.4 TRADUCIDA - EUSKERA - ESPAÑOL:

Si la Administración o la entidad receptora del informe excluye de las recomendaciones recogidas en un informe obligatorio de la Oficina Si la entidad destinataria del informe rechaza las recomendaciones recibidas por la Administración, o la Entidad receptora de los informes recibidos en un informe preceptivo de la propia Oficina, motivará el informe en el informe que se incluirá en el correspondiente expediente de la oficina, que será objeto de publicación en el perfil del contratante y en la Plataforma de Contratación del Sector Público, que deberá publicarse en el informe incluido en el expediente del correspondiente contrato, en el informe añadido al expediente del contrato-contrato, el perfil de contratante, y la plataforma de contratación

del sector público, en el que se añada al expediente de contrato, perfil de reclutador y plataforma de reclutamiento del sector público.

FRASE 2.5: *Contratación, que se basará en el análisis de actuaciones de contratación realizadas por todo el sector público incluyendo todos los poderes adjudicadores y entidades adjudicadoras comprendidas en el sector público estatal, autonómico o local, así como las de otros entes, organismos y entidades pertenecientes a los mismos que no tengan la naturaleza de poderes adjudicadores.*

FRASE 2.5 TRADUCIDA - EUSKERA - ESPAÑOL:

Todos los sectores públicos incluidos todos los poderes adjudicatarios y entidades adjudicatarias incluidos en el sector público estatal, autonómico o local o del sector público local, a partir del análisis de las actividades de contratación realizadas por todo el sector público, incluyendo todos los entes adjudicadores y adjudicadores incluidos por el sector público en su conjunto: todos los sectores públicas incluidos los poderes asignadores e instituciones adjudicadoras incluidas en el Sector Público Estatal, Comunidad Autónoma o Comunidad o Local, incluyendo todas las entidades Adjudicadoras y Adjudicatarias que se integran en el Sector público Estatal, Autonómico o Zona Pública, así como de otras entidades, entidades u organizaciones de otras instituciones, instituciones y organismos que no tengan carácter de poder adjudicatario, o entidades de otros organismos, organismos y organizaciones que carezcan de carácter no adjudicacionarios, o de cualquier otra entidad, entidad u entidades sin carácter del poder adjudicador de aquellas que no tienen carácter poder adjudicacionario.

Ahora, con este análisis, vemos claramente que las repeticiones a la hora de traducir han aumentado considerablemente. Al haber ido en ambas direcciones con nuestros modelos, las repeticiones se han encadenado, obteniendo unos resultados que dejan que desear. La inconsistencia de ambos modelos se ve reflejado en las traducciones.

B.3. Modelo Helsinki-NLP Inglés-Español

En cuanto a los modelos pre-entrenados de ***Helsinki-NLP*** *se realizó un experimento para ver cuánto podía mejorar la BLEU en función del tamaño de datos de entrenamiento. Se trata de un modelo que se especificó en el entorno Institucional, partiendo del* ***Helsinki-NLP*** *Inglés-Español base. Como hemos visto en el Análisis de resultados, es un modelo bastante optimizado, por lo que las mejoras no han sido muy grandes.*

Vamos a realizar varias traducciones para medir tiempos y consumos de gráfica, así como la calidad de las traducciones. Los ejemplos son extraídos de textos del Parlamento Europeo, de los que tenemos las traducciones realizadas por humanos:

FRASE 3.1: *Ladies and gentlemen, before you leave me alone, I would like on behalf of the Bureau, once again, to thank all the Members, all the services, officials, assistants and other co-workers and, if you will allow me - although all the co-workers work for us - perhaps a special mention should go to the language services which help us to understand each other here.*

FRASE 3.1 TRADUCIDA CON MODELO: *Señorías, antes de que me dejen solos, me gustaría, en nombre de la Mesa, dar las gracias una vez más a todos los diputados, a todos los servicios, funcionarios, asistentes y otros compañeros de trabajo y, si me lo permiten -aunque todos los compañeros de trabajo trabajen para nosotros- quizás deba hacerse una mención especial a los servicios lingüísticos que nos ayudan a entendernos aquí.*

FRASE 3.1 TRADUCIDA POR HUMANO: *Señorías, antes de que me dejen solo, yo querría, en nombre de la Mesa, insistir en agradecerles a todas y todos los diputados, a todos los servicios, funcionarios, asistentes y demás colaboradores y, si me lo permiten -aunque todos los colaboradores trabajan para nosotros-, quizá una especial referencia a los servicios lingüísticos que nos ayudan a entendernos aquí.*

FRASE 3.2: *Mr Posselt, I am very pleased, but in any case, I would remind you that, when requesting a procedural motion, you actually have to indicate the Rule to which you are referring. Having said that, Parliament has reached the end of the agenda. The Minutes of the present sitting will be subject to Parliament' s approval at the beginning of the next part-session. Mr Manders has the floor for a procedural motion. Mr President, I would like to take this opportunity to wish you, the Bureau and all Members, a good transition into the new year.*

FRASE 3.2 TRADUCIDA CON MODELO: *Señor Posselt, me congratulo, pero, en todo caso, le recuerdo que, al solicitar una cuestión de orden, tiene que indicar el artículo al que se refiere, es decir, que el Parlamento ha llegado al final del orden del día, que el Acta de la presente sesión estará sujeta a la aprobación del Parlamento al comienzo del próximo período parcial de sesiones, que tiene la palabra el Señor Manders para una cuestión de orden, y aprovecho la ocasión para desearle a usted, a la Mesa y a todos los diputados un buen paso hacia el nuevo año.*

FRASE 3.2 TRADUCIDA POR HUMANO: *Señor Posselt, me alegro mucho. Pero de todas maneras, le recuerdo que, para ser precisos, cuando se pide la palabra para una cuestión de orden hay que hacer referencia al artículo del Reglamento al que se está apelando. Dicho esto, Señorías, el Parlamento ha agotado el orden del día. El Acta de la presente sesión se someterá a la aprobación del Parlamento al comienzo del próximo período parcial de sesiones. Tiene la palabra el Señor Manders para una cuestión de orden. Señor Presidente, quiero utilizar esta oportunidad para desearle a usted, a la Mesa y a todos mis colegas una buena salida y entrada de año.*

FRASE 3.3: *During the debate, I was watching what exactly was written down regarding the vote for the Murphy report on late payment. I would like to ask you to get your department to look at Amendment No 20 again, as I am 99.9 por centum certain that this is not the text we voted on or at any rate not the text which should have been submitted because this is not what the Industry Committee provided. I have asked to be given the floor officially because I fear that we will be unable to reach the departments during the Christmas break and in order to avoid problems during the conciliation procedure. Thank you very much, Mrs Thyssen. We will make the appropriate checks because, evidently, the Minutes have been approved; therefore, there will have to be a technical correction where appropriate.*

FRASE 3.3 TRADUCIDA CON MODELO: *Durante el debate he estado observando lo que estaba escrito exactamente con respecto a la votación del informe Murphy sobre la morosidad, y quisiera pedirle que su departamento vuelva a examinar la enmienda 20, ya que estoy seguro en un 99,9 por ciento de que este no es el texto que votamos o, en cualquier caso, no el texto que se debería haber presentado porque no es el que ha presentado la Comisión de Industria; he pedido que se me conceda la palabra oficialmente porque temo que no podremos llegar a los departamentos durante las vacaciones de Navidad y para evitar problemas durante el procedimiento de conciliación; muchas gracias, señora Thyssen; haremos los controles oportunos porque, evidentemente, se ha aprobado el Acta; por lo tanto, tendrá que haber una corrección*

técnica cuando proceda.

FRASE 3.3 TRADUCIDA POR HUMANO: *Durante el debate, observé lo que se escribió exactamente sobre la votación del Informe Murphy relativo al establecimiento de medidas de lucha contra la morosidad en las operaciones comerciales. Le ruego pida a sus servicios que examinen de nuevo la enmienda 20. Estoy segura en un 99,9 por ciento de que no es el texto que votamos o, en cualquier caso, no el texto que se debía haber presentado, ya que no es el resultado al que llegó la Comisión de Industria. He pedido oficialmente la palabra porque me temo que ya no podamos contactar con los servicios durante las vacaciones de Navidad y, así, también evitar problemas durante el procedimiento de codecisión. Muchas gracias, señora Thyssen. Haremos las comprobaciones oportunas porque, evidentemente, el Acta se ha aprobado; por lo tanto, ha de ser una corrección técnica en su caso.*

FRASE 3.4: *Mr President, I do not know if this is a technical correction, but I have just discovered that I am not included in the Members from Luxembourg in Wednesday' s Minutes, concerning Mrs Palacio Vallelersundi' s report on the verification of credentials. Mrs Reding' s name is there instead. I know that I owe my seat to her being appointed a Commissioner but I do not understand, since I have been a Member of this Parliament since 16 September, why my name does not feature in the Minutes covering the verification of credentials. Would you please rectify this? Mrs Lulling, I cannot rectify this because this report does not affect you. You were elected on 16 September - as you quite rightly said - and this report concerns those who were elected on 13 June. There will therefore be another report, which will, I hope, confirm your mandate. Mr President, as it is now Christmas, I would be grateful if you would allow me to speak for a moment. I would like to thank you and clear up a misunderstanding. The President is entitled to allow an MEP to ask a question of the Commission. I wanted to ask the Commissioner a question and also answer a question that you asked of Mr Mayer. I wanted to say that in the spring there will be a large beer-tasting session for Bavarian beer in the Parliament courtyard here in Strasbourg.*

FRASE 3.4 TRADUCIDA CON MODELO: *Señor Presidente, no sé si se trata de una corrección técnica, pero acabo de descubrir que no estoy incluido en el Acta de los diputados luxemburgueses del miércoles, en relación con el informe de la Señora Palacio Vallelersundi sobre la verificación de credenciales. Sé que mi escaño se debe a que ha sido nombrada Comisaria, pero no entiendo, puesto que soy diputada a este Parlamento desde el 16 de septiembre, por qué mi nombre no figura en el Acta relativa a la verificación de credenciales, por favor, rectifique esto, señora Lulling, no*

puedo rectificarlo porque este informe no le afecta, usted fue elegido el 16 de septiembre -como muy bien ha dicho- y este informe se refiere a los que fueron elegidos el 13 de junio, por lo que habrá otro informe que, espero, confirmará su mandato. Me gustaría darle las gracias y aclarar un malentendido: el Presidente tiene derecho a permitir que un diputado al Parlamento Europeo formule una pregunta a la Comisión. Quería hacer una pregunta al Comisario y también responder a una pregunta que usted ha formulado al Señor Mayer: quería decir que en primavera se celebrará una gran sesión de degustación de cerveza bávara en el patio del Parlamento aquí en Estrasburgo.

FRASE 3.4 TRADUCIDA POR HUMANO: *Señor Presidente, ignoro si se trata de una corrección técnica pero acabo de descubrir que no figuro entre los diputados luxemburgueses en el Acta del miércoles, relativa al informe de la Señora Palacio Vallelersundi sobre la verificación de poderes. Se cita a la Señora Reding. Sé que debo mi escaño a su nombramiento como Comisaria, pero no comprendo por qué aun siendo diputada desde el 16 de septiembre, no figuro en el Acta sobre verificación de poderes. ¿Puede, por favor, rectificarlo? Señora Lulling, no puedo rectificar porque no es usted objeto de ese informe. Ha sido elegida el 16 de septiembre -como dice correctamente- y este informe concierne a los elegidos el 13 de junio. Por consiguiente, habrá otro informe que, espero, confirme su mandato. Señor Presidente, puesto que estamos en Navidad, quisiera que me concediese usted un poco de tiempo. Quiero darle las gracias a usted y explicar un malentendido. El Presidente tiene el derecho de autorizar a un diputado una pregunta a la Comisión. Yo quisiera efectuar una pregunta a la Señora Comisaria y, con independencia de esto, responder a una pregunta que usted dirigió a nuestro colega, el Señor Mayer: quisiera decir que en primavera tendrá lugar aquí, en el patio del Parlamento, en Estrasburgo, una gran degustación de cerveza bávara.*

FRASE 3.5: *Following the vote on the amendments. Mr President, after the voting, I would like to raise a point of order concerning the texts adopted yesterday. If you would allow me, I would like to take up a few minutes after the votes. You may do so. Parliament approved the legislative resolution EXPLANATIONS OF VOTE. Mr President, I would like to say how pleased I am to give this last explanation of vote of 1999 on the Savary report, which I voted for. I am very much in favour of this European Community initiative which aims to grant practical aid to the most deserving regions, such as Tajikistan. I would like, as Mr Blokland said in his statement, to say to Commissioner Wallstrm - who is disturbing the sleep of Italian, Greek and Spanish pensioners because of superannuated cars which have to be quickly taken off the roads - that I would appreciate verification as to how the aid is distributed once it has been*

granted to Tajikistan and other States. I would be pleased if it was given to people who need it, such as pensioners. Mr President, despite having some serious concerns, I voted in favour of the financial aid package because I see in it an opportunity to establish a market economy, democracy and peace using financial incentives, of the kind put forward by Mr Karas, rather than force. I would like to extend my heartfelt thanks to all those fellow MEPs who gave me their support in this and would like to wish you all a merry Christmas and Happy New Year! Mr President, very briefly on a point of order regarding the texts adopted yesterday.

FRASE 3.5 TRADUCIDA CON MODELO:

Después de la votación sobre las enmiendas. Señor Presidente, después de la votación, quisiera plantear una cuestión de orden relativa a los textos aprobados ayer; si me lo permiten, me gustaría ocuparme de algunos minutos después de las votaciones; pueden ustedes hacerlo; el Parlamento aprobó la resolución legislativa EXPLICACIONES DE VOTO; señor Presidente, quisiera decir cuánto me complace dar esta última explicación de voto de 1999 sobre el informe Savary, que he votado a favor. Estoy totalmente a favor de esta iniciativa de la Comunidad Europea que pretende conceder ayuda práctica a las regiones más meritorias, como Tayikistán. Como ha dicho el Señor Blokland en su declaración, quisiera decir a la Comisaria Wallstrm -que está perturbando el sueño de los pensionistas italianos, griegos y españoles debido a los coches sobranuados que hay que sacar rápidamente de las carreteras- que agradecería que se verificara cómo se distribuye la ayuda una vez que se ha concedido a Tayikistán y a otros Estados. Señor Presidente, a pesar de tener algunas preocupaciones serias, he votado a favor del paquete de ayuda financiera porque veo en él una oportunidad para establecer una economía de mercado, democracia y paz con incentivos financieros, del tipo propuesto por el señor Karas, en lugar de la fuerza. Quiero expresar mi más sincero agradecimiento a todos los colegas que me han apoyado en este asunto y desearles a todos una feliz Navidad y un feliz año nuevo, señor Presidente, muy brevemente sobre una cuestión de orden relativa a los textos aprobados ayer.

FRASE 3.5 TRADUCIDA POR HUMANO: *Además, a la luz de las actuales dificultades en el Cáucaso, deberíamos enviar señales políticas acerca de nuestra voluntad de continuar apoyando sus grandes esfuerzos para hacer reformas y para alcanzar estabilidad y democracia. Tras la votación de las enmiendas. Señor Presidente, tras las votaciones, todavía tengo una cuestión de orden relativa a los textos aprobados ayer. Si usted me lo permite, pediré la palabra un instante tras las votaciones. De acuerdo. El Parlamento aprueba la resolución legislativa EXPLICACIONES DE VOTO. Señor Presidente, siento una gran satisfacción formulando esta última explicación de voto del año 1999 a favor del informe del Señor Savary que he votado.*

Respaldo esta iniciativa de la Comunidad Europea por la que se concede una ayuda concreta a las regiones más meritorias como Tayikistán. Al igual que lo ha hecho el Señor Blokland en su intervención, quisiera decirle a la Comisaria Señora Wallstrm -que quita el sueño a los pensionistas italianos, griegos y españoles por los vehículos antiguos que han de ser eliminados rápidamente- que agradecería que se controlara el destino de las ayudas concedidas a Tayikistán y a otros Estados. Me gustaría que se destinaran también a las personas que las necesitan como, por ejemplo, los pensionistas. Distinguido señor Presidente, a pesar de grandes reparos, he votado a favor de la ayuda financiera, porque veo una posibilidad de establecer la economía de mercado democracia y la paz mediante estímulos financieros, como ha expuesto el Señor Karas, y no a través de la violencia. Quisiera dar encarecidamente las gracias a todos los colegas que me han apoyado en esto, les deseo una feliz Navidad y un próspero año nuevo. Señor Presidente, brevemente, sobre los textos aprobados ayer.

Tras analizar todas y cada una de las traducciones, vemos que son muy precisas, pues con todas prácticamente idénticas a la original. Esto se debe a ser un modelo pre-entrenado, y, tras haberlo especializado en Instituciones, la calidad de las traducciones mejora drásticamente. Pasamos a ver la tabla B.2 que resume las medidas obtenidas tras la traducción de estas secuencias.

Tabla B.2: Tiempos y consumos de las frases analizadas, con ***Helsinki-NLP*** Inglés-Español

Frase	Tiempo de ejecución medio (s)	Consumo medio (MB)	Número de palabras
Frase 1	1,6	296	62
Frase 2	2,08	314	96
Frase 3	2,52	358	141
Frase 4	0,9	46	237
Frase 5	0,77	46	277

*Como ocurría en el caso del **mT5**, al fijar el número máximo de palabras por frase a 200, los resultados al truncar son muy favorables, pues los tiempos y consumos son muy bajos. Además, como ocurre con los modelos de **mT5** entrenados, la calidad de traducción es mucho mejor cuando las secuencias de entrada son cortas, pues la gran mayoría de ejemplos de entrenamiento tenían pocas palabras.*

*Si comparamos estos resultados con los de la tabla B.1, donde se ha utilizado un modelo **mT5**, este modelo es mucho más ligero, pues traduce más rápido y consume menor gráfica al traducir.*

*Concluimos con que los modelos de **Helsinki-NLP** analizados presentan muy*

buenos resultados, por lo que se confirma que están muy bien optimizados.

Anexos C

Redes Neuronales Artificiales y Transformers

C.1. Redes Neuronales Artificiales

Las redes neuronales artificiales son un modelo matemático inspirado en cierta medida en el funcionamiento de las redes de neuronas biológicas humanas, es decir, en el funcionamiento del cerebro humano. Del mismo modo que nuestro cerebro está compuesto por neuronas interconectadas entre sí, una red neuronal artificial está formada por neuronas artificiales conectadas entre sí y agrupadas en diferentes niveles que denominamos capa. El principal objetivo que tienen es la transmisión de información de unas a otras desde la entrada, hasta generar una salida.

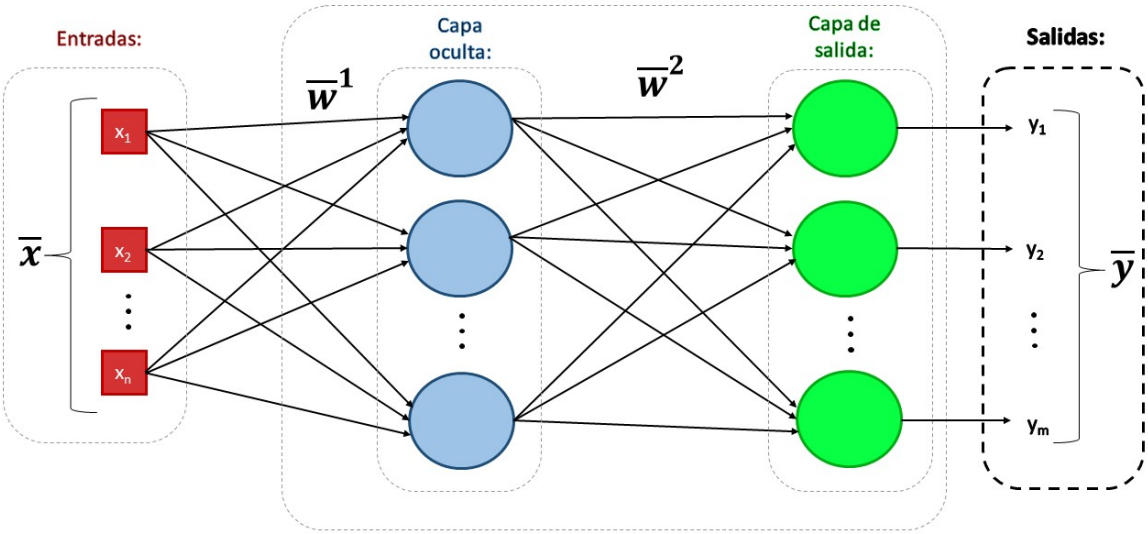

Figura C.1: Estructura red neuronal artificial

La estructura de una red neuronal artificial se puede observar en la figura C.1, donde

las neuronas de la primera capa reciben como entrada los datos reales que alimentan a la red neuronal, conocida como la capa de entrada. La salida de la última capa es el resultado visible de la red, por lo tanto, la última capa se conoce como la capa de salida. Las capas que se sitúan entre la capa de entrada y la capa de salida se conocen como capas ocultas, ya que desconocemos tanto los valores de entrada como los de salida. Se resume la función de transferencia de una red neuronal en la ecuación C.1, donde se obtiene la salida "y" de una neurona.

$$y = b + \sum_{i=1}^{N} w_i x_i \tag{C.1}$$

Cada neurona posee un conjunto de valores numéricos llamados pesos, uno por cada entrada, con los que modifican las entradas recibidas, dando lugar a unas salidas correspondientes, que continuarán su camino por la red. En la ecuación C.1, la neurona transforma a través de la función de activación la suma del sesgo con el producto vectorial de la entrada con los pesos de la capa i-ésima.

La función de activación ***sigmoide****, que podemos observar su función de transferencia en la figura C.2, es la función de activación más antigua y popular. Es capaz de comprimir la entrada en el rango de 0 a 1; es decir, para valores negativos grandes de "z", el término e^{-z} en el denominador crece exponencialmente, y $\sigma(z)$ se aproxima a 0. Al contrario, valores positivos grandes de "z" reducen el término e^{-z} hacia 0, y $\sigma(z)$ se aproxima a 1.*

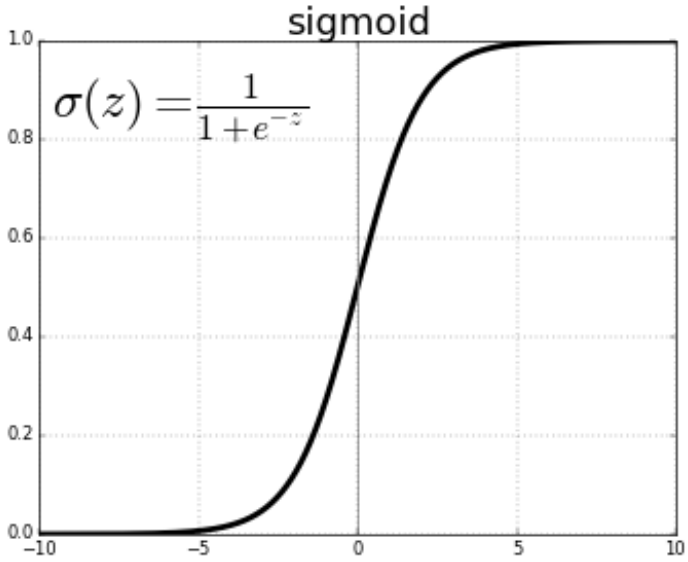

Figura C.2: Función de activación *sigmoide*

Las redes son aproximadores univerales, es decir, al aumentar el número de capas

presentan un mayor número de parámetros libres, capaces de representar con más precisión funciones de mayor dificultad.

En cuanto a las propiedades para clasificar las redes neuronales artificiales, destacamos la topología, es decir, el esquema de conexión que forman las neuronas. En ***NLP (Natural Language Proccessing)****, la gran mayoría de redes se pueden clasificar en dos grandes grupos: las redes recurrentes* ***RNN (Recurrent Neural Network)*** *[31] y las redes multicapa* ***feed-forward****.*

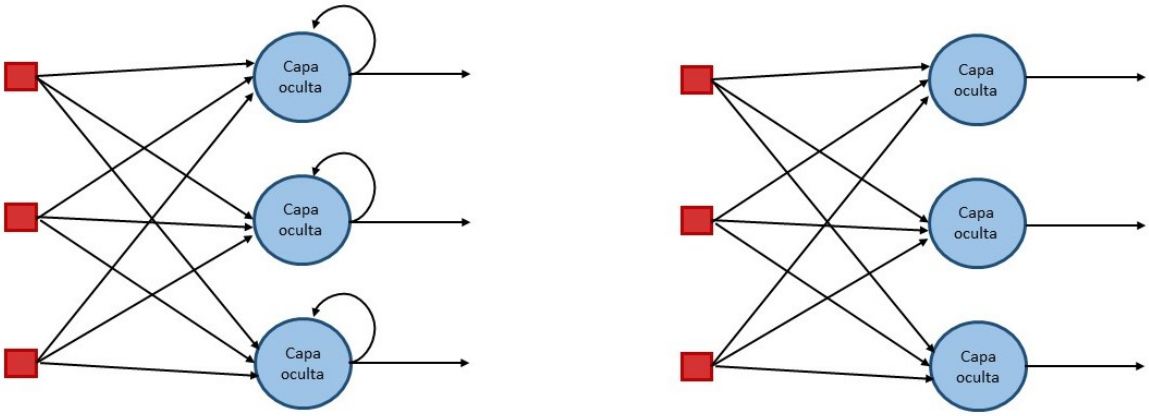

Figura C.3: Comparación entre redes neuronales recurrentes y redes neuronales feed-forward

Se puede observar en la figura C.3 cada una de las estructuras de ambos tipos de redes, donde se tiene a la izquierda las RNN y a la derecha las feed-forward.

Las primeras en aparecer fueron las redes ***feed-forward*** *[27] [31]. Su principal característica es no tener ciclos, pues están formadas por capas que se conectan en una única dirección. Asimismo, el estado de una red es totalmente independiente del estado anterior, por ello, se denominan redes estáticas. Este tipo de redes no obtuvieron resultados destacables en el ámbito de la traducción en aquellos momentos.*

A consecuencia de ello, surgieron las redes recurrentes, o también llamadas ***Recurrent Neural Network (RNN)****. Su funcionamiento es dinámico, basado en el uso de datos secuenciales o datos de series de tiempos. Es decir, toman información de entradas anteriores para influir en la entrada y salida actuales. Por lo tanto, tenemos dependencia temporal y conseguimos añadir un contexto a las secuencias a lo largo del tiempo, a diferencia de las* ***feed-forward****, que los estados son independientes entre sí. Este tipo de redes, al contrario que las anteriores, presentaron un gran progreso y fueron muy importantes en el desarrollo del* ***Deep Learning****.*

C.1.1. Redes Neuronales Recurrentes (RNN)

*La historia de las **RNN** se remonta a la década de los años 80, cuando surgieron dichas redes. Han sido muy difíciles de entrenar por sus requerimientos en computación y hasta la llegada de los avances de estos últimos años, no se han vuelto más accesibles y popularizado su uso por la industria. Están formadas por modelos de neuronas diferentes a los existentes con anterioridad, ya que presentan una función de activación que no solo actúa en una dirección, sino que incluye conexiones en ambas direcciones. Esto supone que cada una de las neuronas presentes en esta red recibe dos entradas en cada instante de tiempo: la entrada correspondiente de la capa anterior (presente también en las redes **feed-forward**) y la salida del instante anterior de la misma capa (nueva incorporación).*

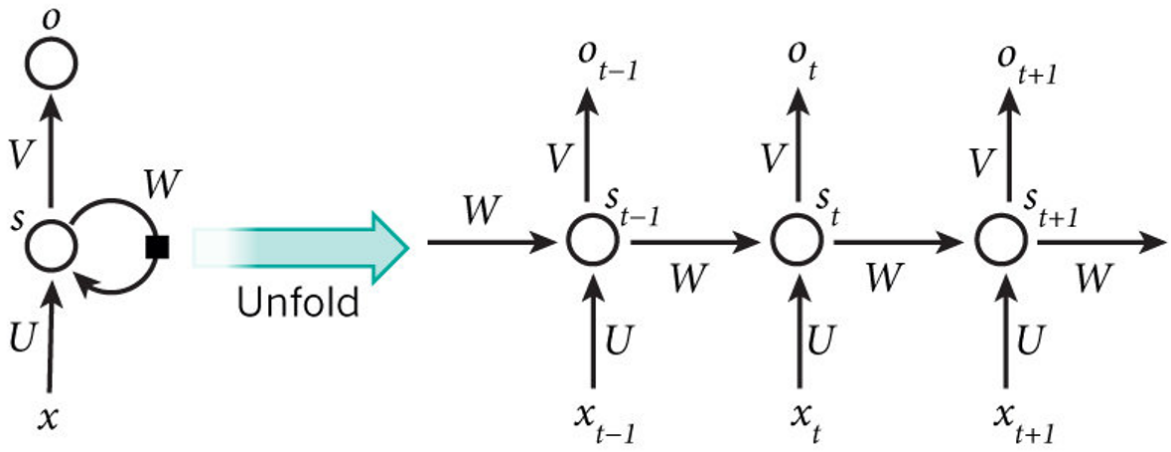

Figura C.4: Red Neuronal Recurrente arquitectura interna

*En la figura C.4 se puede ver el esquema interno de este tipo de redes en función del tiempo, donde tenemos una **RNN** desplegada.*

Se puede asegurar que dichas redes cuentan con cierta memoria, ya que contienen información de instantes anteriores. Gracias a esto, son capaces de recordar información importante que ha sucedido con anterioridad, lo que les permite predecir información con mayor exactitud. Son muy útiles y eficaces para tareas de aprendizaje automático, pues dichas tareas se basan en la utilización de datos secuenciales.

A modo de ejemplo, supongamos que se introduce en la red una frase compuesta por varias palabras. La red procesará dicha frase palabra a palabra, recurrentemente, de forma que va a recordar las palabras anteriores a la entrada actual. De esta manera, podemos observar cómo las secuencias de datos contienen información trascendental para saber lo que viene a continuación, siendo ésta la gran ventaja de estas redes frente a las llamadas redes estáticas, las cuales no guarda información previa.

A pesar de presentar grandes ventajas, estas redes también tienen alguna serie de

inconvenientes, que son los siguientes:

- No es sencillo acceder a la información de los estados muy antigüos, por lo que, a veces, las predicciones no son precisas. Esto ocurre cuando hay un espacio muy grande entre la predicción y el contexto, porque dicha memoria es sólo factible en un pasado cercano.

- Los tiempos de procesado requeridos son muy elevados debido a la complejidad interna de la red.

*Con el objetivo de solventar estos problemas y obtener mejores resultados, apareció el modelo denominado **Transformer** [7]. Este modelo se comenta detenidamente en el Anexo C.*

C.2. Transformer

*En un paper de Google a finales del año 2017 se presentó la arquitectura del **Transformer** [7], un modelo que tenía como principal mejora un gran avance en el uso del mecanismo de atención.*

C.2.1. Funcionamiento

*Las capas de atención mencionadas codifican cada palabra de una frase en función del resto de la secuencia, permitiendo así introducir el contexto en la representación matemática del texto. La idea clave del **Transformer** es gestionar completamente las dependencias entre la entrada y la salida prescindiendo totalmente de técnicas de recurrencia y, en algunos casos, convolucionales.*

En la figura C.5 se muestra el esquema de dicho modelo.

*La arquitectura se basa en un **encoder-decoder**, es decir, está formada por un codificador, situado a la izquierda de la imagen, y un decodificador, situado a la derecha. El encoder es el encargado de procesar y analizar el contexto de la secuencia de entrada, mientras que el decoder genera la secuencia de salida a partir de ese contexto.*

Pasamos ahora a analizar en detalle la arquitectura presentada en la figura C.5:

*- En el encoder y decoder, ambos bloques van precedidos de un diccionario de **embeddings**, los cuales se encargan de convertir el texto de entrada en una serie de vectores, o **tokens**, para así tener una representación numérica de cada una de las secuencias de entrada. Además, es esperable que después del proceso de entrenamiento las palabras con sentido semántico similar estén cercanas en dicho espacio vectorial [30].*

*- Tras la realización del **embedding**, es necesario hacer una codificación posicional,*

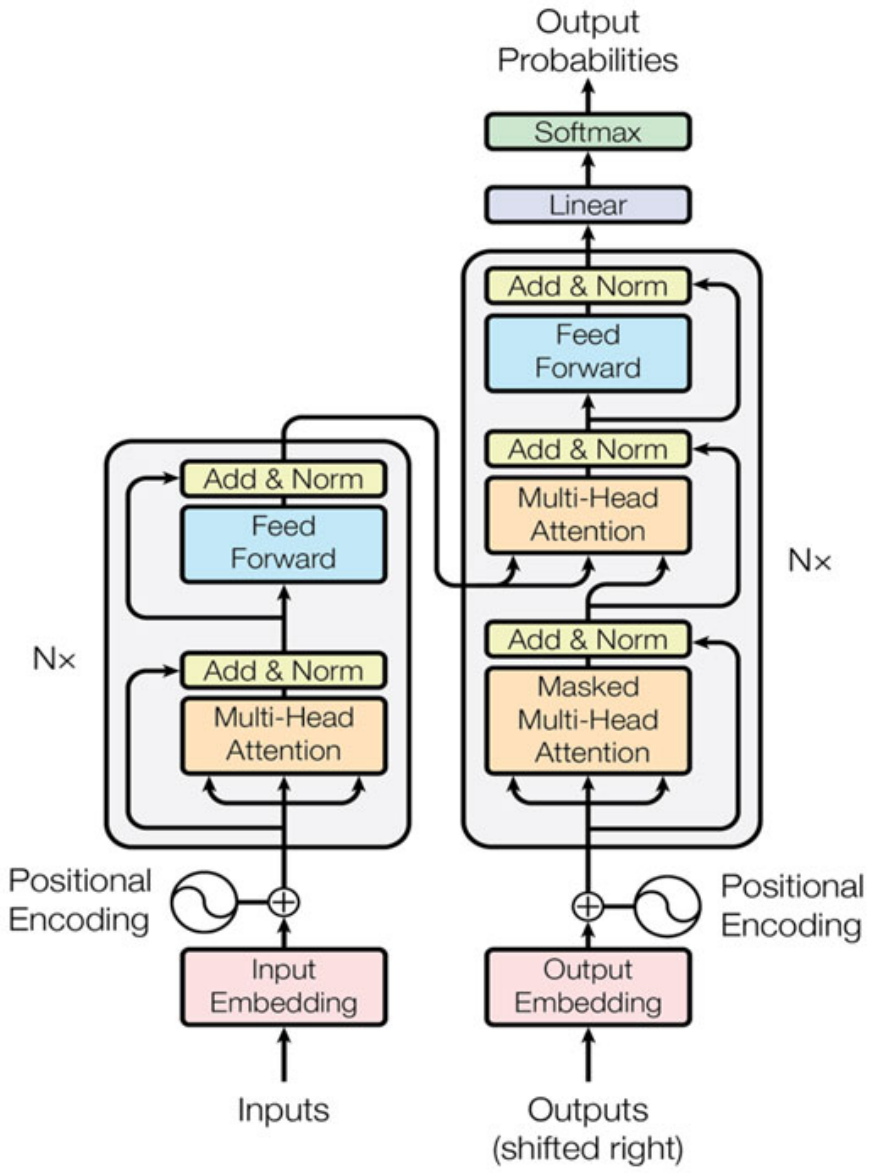

Figura C.5: Arquitectura del *Transformer* [7]

ya la secuencia de entrada se procesa en paralelo. Además, el orden y posición de las palabras es algo esencial para comprender cualquier idioma. Este codificador genera una serie de vectores que se sumarán a los tokens, y que indican la posición relativa de cada token dentro de la secuencia. Para esto se usan funciones senoidales para las posiciones pares, y cosenoidales para las impares, con lo que cada vector generado tendrá un patrón numérico único con la información de la posición. Los vectores resultantes ingresarán a la etapa de codificación, que se encarga de extraer la información mas relevante de la secuencia en su idioma original.

Ahora, tras haber acabado el proceso anterior a la entrada del codificador y decodificador, procedemos a estudiar la estructura interna de ambos, de forma más

detallada.

El bloque encoder esta formado por 6 bloques o capas de estructura idéntica, como se puede observar en la figura C.6.

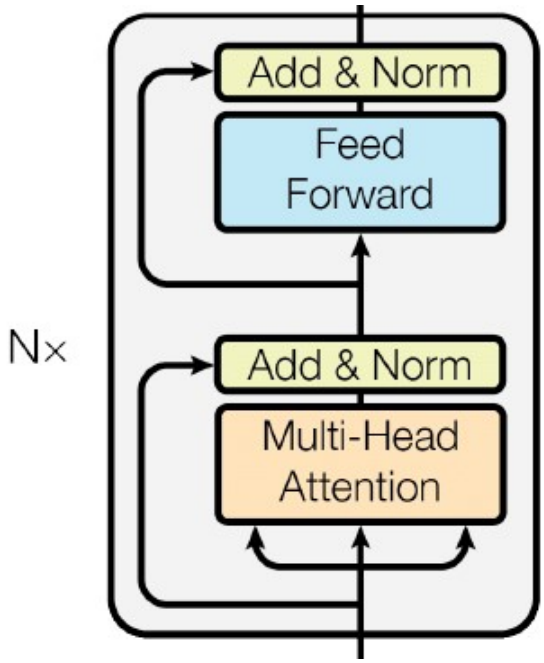

Figura C.6: Estructura interna del encoder [7]

Cada codificador está a su vez formado por 4 subbloques: uno de auto-atención, dos de conexión residual y otro que contiene una red neuronal feedforward (retroalimentación). Veamos qué ocurre dentro de cada subbloque:

- El subbloque de auto-atención es, quizás, el más importante de toda la red, pues se encarga de analizar la totalidad de la secuencia de entrada (recordemos que la red la procesa de manera simultánea) y de encontrar relaciones entre varias palabras de esta secuencia. Es la gran novedad que incorporan los ***Transformers*** *al mundo de las redes neuronales artificiales y gracias a la cual se consiguen las grandes mejoras respecto a modelos creados con anterioridad. Así, lo que hace el bloque atencional es expresar numéricamente las relaciones que existen a diferentes niveles dentro de la secuencia, y luego codifica cada una de ellas con esta información del contexto, indicando así cuáles son los elementos del texto a los que se deben prestar más atención al momento de hacer la traducción. Esta es precisamente la manera como las redes transformer "comprenden" este contexto para codificar adecuadamente cada palabra.*

Para entender mejor este novedoso concepto, se expone en la figura C.7 un ejemplo de cómo opera esta capa interna del codificador:

Se muestra, mediante cabeceras de distintos colores, las dependencias de la palabra 'it' en inglés. Así es como se consigue tener conocimiento de cuáles son los elementos

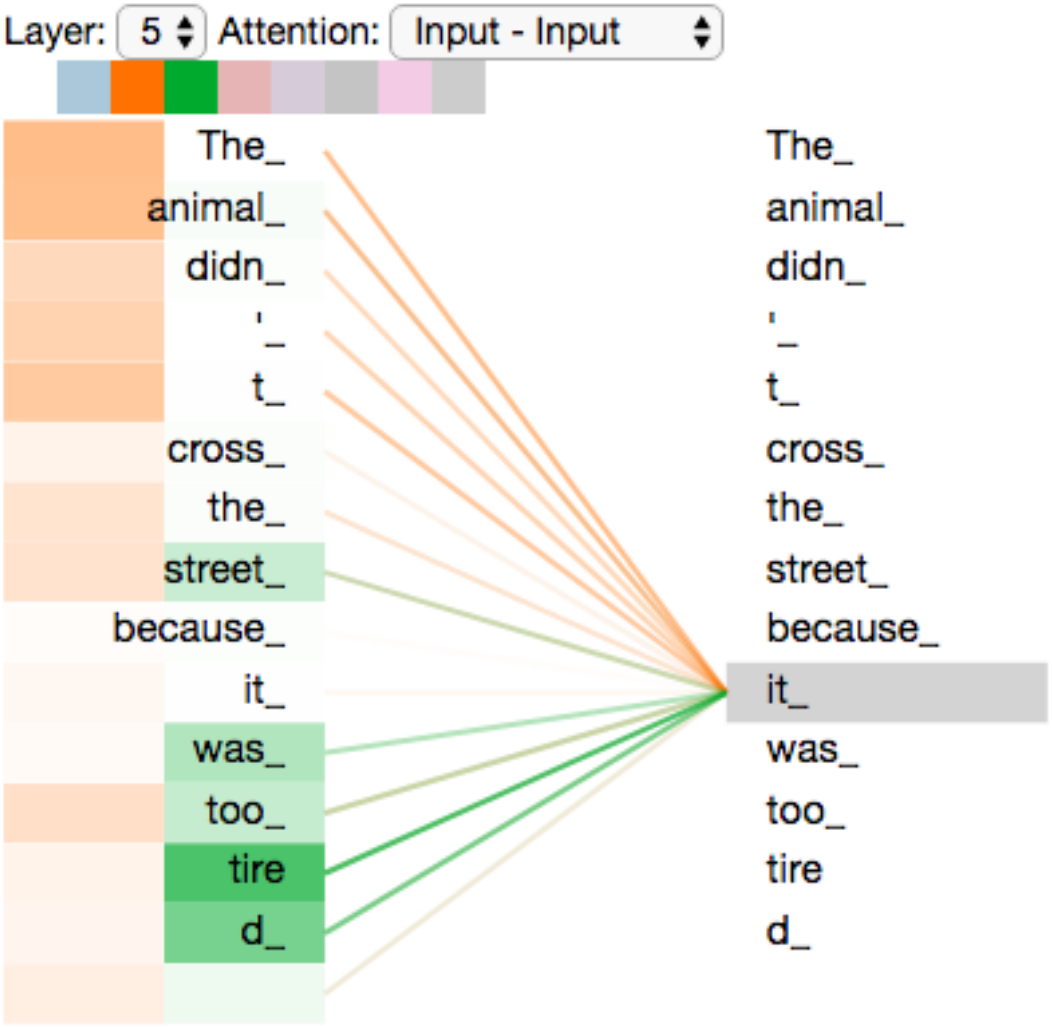

Figura C.7: Ejemplo del subbloque de auto-atención [32]

del texto a los que se debe prestar mayor atención a la hora de predecir las palabras que conformaran la traducción.

Adentrándonos un poco más en el subbloque de auto-atención, podemos ver configuración interna en la figura C.8. Para lograr dicha comprensión del texto, en primer lugar los tokens se llevan simultáneamente a tres pequeñas redes neuronales, entrenadas para calcular los vectores "query", "key" y "value". Estos vectores son simplemente tres representaciones alternativas de los tokens originales:

- Q: se trata de la matriz ***query matrix****, donde cada columna contiene el vector o embedding de los tókens de salida que tienen que predecir el modelo.*

- K: las ***keys matrix****, cada columna son los embeddings de entrada, es decir, el resto de la secuencia.*

- V: Almacena los valores ***(values matrix)*** *vectoriales de una palabra procesada en un determinado instante temporal. En el caso del encoder, los vectores V y K son*

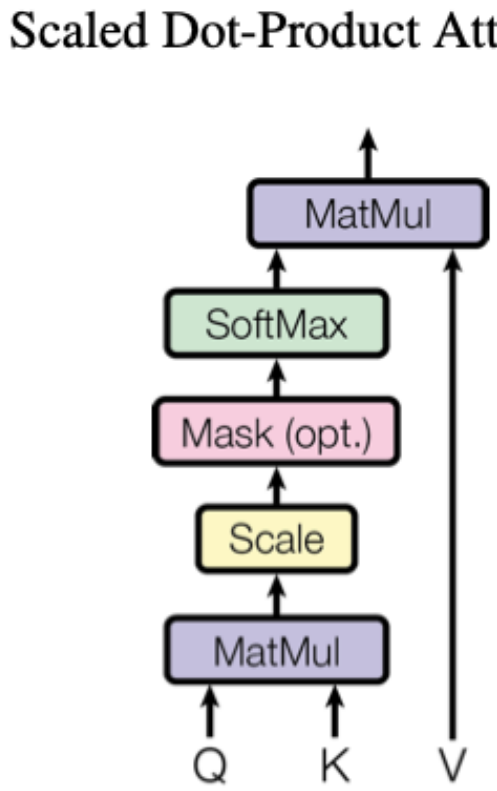

Figura C.8: Configuración interna subbloque auto-atención [7]

idénticos.

Una vez obtenidos los parámetros de entrada, se dispone a sacar una matriz puntuación capaz de medir el grado de asociación entre pares de palabras. Para ello, se toma el ***embedding*** *de cada token y se compara con cada uno de los* ***embedding keys*** *existentes. El resultado es la matriz puntuación, la cual se calcula multiplicando ambas matrices.*

Posteriormente, es necesario escalar la matriz puntación, dividiéndola por el tamaño de cada vector e introduciéndola en una función softmax [33]. Tras esta función, cada salida es una suma ponderada de los vectores de entrada V, haciendo que la suma sea 1. Por lo tanto, representa cada puntuación como un valor entre 0 y 1. Los tókens cercanos a 1 serán relevantes y se prestará más atención.

Para finalizar, se multiplica la matriz de puntuación por V, el vector de la palabra con la que estamos trabajando, para reducir la importancia de palabras no relevantes y quedarnos solo con las que nos importan. Así, se consigue como resultado esta misma matriz, pero ponderada.

La fórmula que representa todos los cálculos comentados es la siguiente:

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

Figura C.9: Mecanismo de Auto-atención [34]

En resumidas cuentas, el bloque de auto-atención contiene una combinación lineal de los elementos en la entrada que tienen una mayor relevancia.

- El siguiente bloque tras el de auto-atención, como podemos ver en la figura C.6, es un bloque residual (Add and Norm, Agregar y Normalizar) [35]. Es capaz de evitar la degradación de la información, pues toma tanto la entrada como la salida del bloque de auto-atención, ya que si únicamente tomara la salida, la información progresivamente se vería degradada. Ambos datos se suman y se normalizan, de forma que se obtiene la escala adecuada requerida por el siguiente bloque.

El bloque de auto-atención es capaz de encontrar asociaciones entre palabras. Sin embargo, para detectar asociaciones entre grupos de palabras, un solo bloque no es capaz, por lo que se necesitan múltiples de ellos. Esto es el llamado ***MultiHead Layer****, cuya base es proyectar linealmente K, Q y V en h espacios lineales. Esto permite que cada cabeza se centre en aspectos diferentes, para después concatenar los resultados. El tener varios subespacios y, por lo tanto, varias representaciones de importancia de cada palabra, permite que la propia palabra no sea la dominante en el contexto.*

- Consecuentemente, como se muestra en la figura C.6, se incluye más adelante una red neuronal o ***feed-forward****, que es la encargada de procesar todos los datos en paralelo de las distintas capas y consolidarlos en una única salida final. Además, tanto la entrada como la salida de esta red serán llevadas a otro bloque residual que realizará el mismo objetivo que el bloque idéntico anterior.*

Así quedaría explicado, al completo, el funcionamiento del codificador. Este proceso se repite para los codificadores restantes, que son idénticos en estructura al codificador que acabamos de analizar.

Ahora, nos enfocamos en el decodificador, el siguiente bloque importante de la red Transformer. El bloque decoder cuenta al igual que el encoder con 6 decodificadores, todos ellos conectados a los codificadores, por lo que tendrán acceso a la información de atención codificada en la entrada, en el idioma original, para poder realizar la traducción. Su estructura es la siguiente:

Como podemos ver en la figura C.10, cada decodificador es similar a los bloques de codificación que vimos anteriormente: cuenta con bloques de auto-atención, residuales y redes neuronales que tienen la misma estructura de los codificadores. Sin embargo

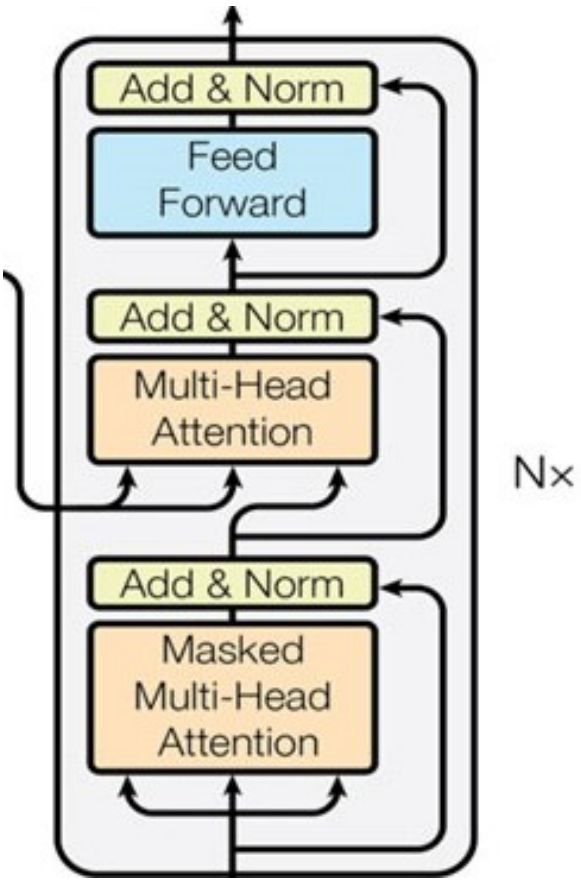

Figura C.10: Estructura interna del decodificador [7]

tienen un bloque de auto-atención con enmascaramiento y un bloque residual adicional.

Procedemos a explicar paso a paso el funcionamiento del decodificador.

- El primer subbloque con el que empieza la decodificación es el bloque de auto-atención con enmascaramiento. Es muy parecido al bloque de auto-atención explicado en el codificador, pero cuenta con una diferencia importante. ***Multi-Head Attention*** *es un subbloque que realiza el proceso de auto-atención en paralelo "h" veces proyectando las matrices Q, K y V y, tras eso, se obtienen "h" salidas, las cuales se concatenan en un vector y se vuelven a proyectar para obtener el resultado final. La idea detrás del* ***Multi-Head Attention*** *es permitir que la función de atención extraiga información de diferentes subespacios de representación, lo que, de otro modo, no sería posible con una sola capa de auto-atención.*

- Al igual que con el codificador, en este caso también se emplean múltiples bloques atencionales para detectar relaciones a diferentes niveles. Todos los bloques residuales, así como la red neuronal de este decodificador funcionan de forma idéntica a como ocurría en los codificadores, salvo por la gestión de las matrices K, Q, V y que la entrada del decodificador son los símbolos decodificados anteriormente.

- Por lo tanto, nos enfocaremos ahora en el bloque de auto-atención, que en este

caso tiene la misma estructura, pero un funcionamiento ligeramente diferente al del codificador. Al estar en el decodificador, donde se genera la salida, es necesario que la atención se enfoque tanto en la secuencia original como en la secuencia de salida. Como se puede observar en la figura C.5, una de las entradas al bloque de auto-atención es la salida del propio codificador, utilizándose como entrada las matrices ***query*** *y* ***keys*** *nombradas anteriormente. Por otro lado, el vector* ***values*** *usa como entrada el dato proveniente del bloque residual anterior. De esta manera, el codificador indica al decodificador a qué elementos debe prestar más atención a la hora de generar la secuencia de salida. Del mismo modo que el codificador, es* ***Multi-Head****, es decir, cuenta con varias capas, aunque en el decodificador se usa una máscara causal. Este bloque se replica un total de seis veces, y al final genera un vector con cantidades numéricas.*

- Finalmente, tras pasar toda la información por los subbloques explicados y vistos en la arquitectura, lo único que falta es convertir la secuencia numérica en una palabra. Para ello, se usa en primer lugar la capa lineal, que es simplemente una red neuronal que toma el vector producido por el decodificador y lo transforma en un vector mucho más grande, del tamaño del vocabulario que ha aprendido el traductor.

En segundo lugar, tras la capa lineal, se pasa por una capa ***softmax*** *[33]. Dicha capa toma cada elemento de este vector y lo convierte en una probabilidad, todas con valores positivos entre 0 y 1. La posición con la probabilidad más alta será seleccionada y la palabra asociada con dicha posición será precisamente la salida del modelo en ese instante de tiempo. Este proceso se repite hasta generar la secuencia de salida.*

C.2.2. Estructuras basadas en Transformers

A parte de la arquitectura genérica del modelo ***Transformer****, aparecen otros tipos de estructuras basadas en ella. Las estructuras que se van a explicar a continuación están preentrenadas con texto plano, a diferencia de la estructura principal de Transformers, que sirve para entrenar modelos desde cero.*

Codificadores *En ocasiones no es necesaria la estructura completa de un modelo* ***Transformer*** *vista anteriormente, sino que, con solo una parte de ella, somos capaces de realizar ciertas tareas más sencillas, pero igual de útiles. Por ejemplo, algunas tareas pueden ser: Clasificación de oraciones, reconocimiento de entidades nombradas o respuesta a preguntas extractivas. Este tipo de tareas son denominadas del tipo* ***Language Understanding****.*

En estos casos se busca un modelo capaz solo de codificar el texto, en vez de realizar tareas más complejas. Para ello, aparecen modelos que son el resultado de tomar una

*red de **Transformer** y quedarnos únicamente con el bloque de codificación.*

*Uno de los modelos más destacados y conocidos que nos ofrece estas características es el modelo **BERT** [36]. Fue desarrollado en octubre de 2019 y es el acrónimo de 'Bidirectional Encoder Representations from Transformers'. Fue creado por Google y cabe tener en cuenta que, antes de su creación, Google no tenía respuesta para el 15 % de las búsquedas realizadas.*

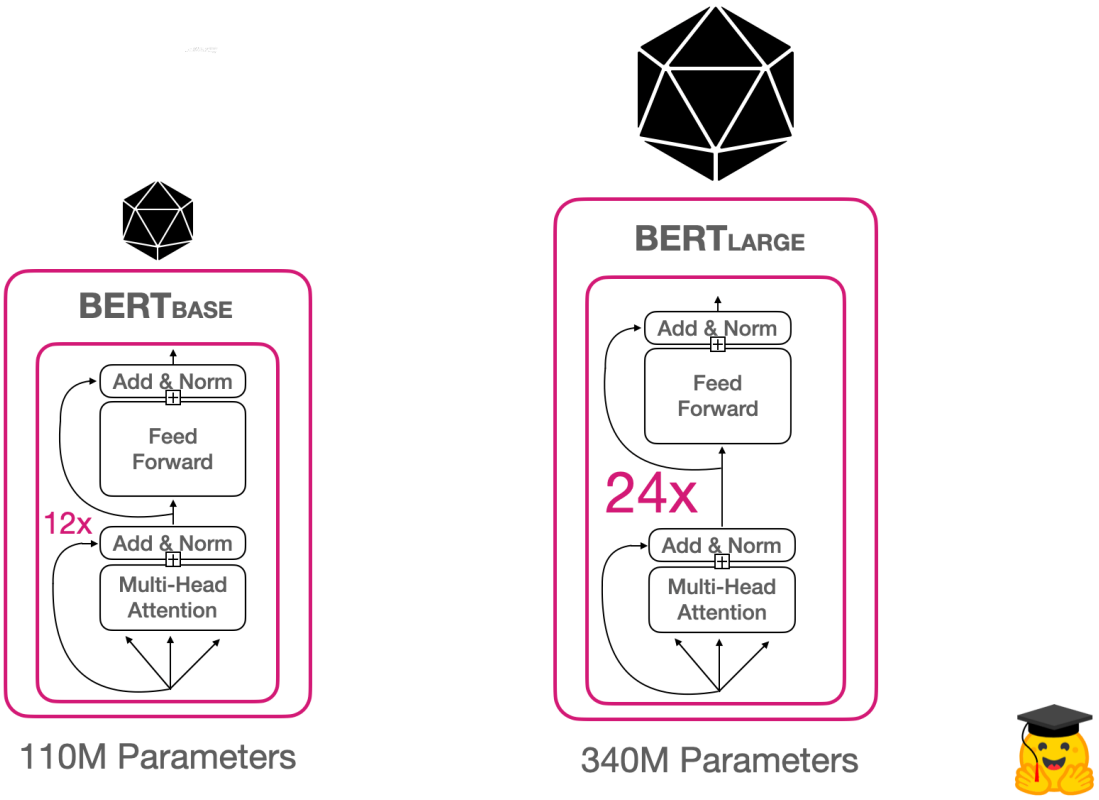

Figura C.11: Estructura y tamaños del modelo ***BERT*** [37]

*Visualizando la arquitectura del modelo **BERT** en la figura C.11, comprobamos que se basa en el codificador del modelo **Transformer** explicado en anteriormente. Dependiendo del número de parámetros, se puede clasificar en dos tipos de modelos: BASE y LARGE.*

*La característica más importante de este modelo es que posee bidireccionalidad, es decir, es capaz de analizar una oración en varias direcciones, siendo estas el futuro y el pasado de cada palabra. Esto permite al modelo entender en profundidad el contexto de la secuencia y la temática de toda la frase. Por todos los aspectos mencionados, **BERT** generó una revolución dentro del procesado del lenguaje natural (**NLP**).*

Decodificadores *Otros tipos de estructuras basadas en el modelo **Transformers** que prescinden de parte de su arquitectura son los llamados modelos de decodificador. Dichos tipos de modelos son capaces de realizar tareas de **Language Generation***

como: Generación de textos, creación de lenguajes de programación o Chatbots. Cuando se habla de generación de texto, se habla de introducir al modelo dos o tres palabras iniciales y a partir de ellas el modelo generar un texto.

Este tipo de tarea genera una secuencia de salida sin depender de ninguna secuencia de entrada. Ocurre lo mismo en el caso de los Chatbots. Por consiguiente, este tipo de modelos no necesitan un codificador que procese una entrada, sino que su arquitectura contiene única y exclusivamente decodificadores.

Los modelos que más han revolucionado el mundo de la inteligencia artificial en los últimos años y que están basados en este tipo de estructuras son los llamados ***GPT*** *(Generative Pre-trained Transformer) [38]. La principal característica que presentan estos modelos es la gran cantidad de parámetros que almacena, teniendo* ***GPT-3*** *[39] hasta 175.000 millones. Esto dota al modelo de una gran capacidad de almacenar y recuperar patrones vistos en el entrenamiento, y algunos patrones básicos de razonamiento aprendidos de los datos.*

Es importante saber que este tipo de modelos están en continuo desarrollo y este es solo el principio del largo camino que van a recorrer.

Secuencia a secuencia (Seq2Seq). *Este tipo de modelos ya apareció con las redes neuronales, aunque, a diferencia de los dos anteriores, necesitan la arquitectura Transformer al completo, tanto codificador como decodificador. Esto es así porque las tareas que realizan se basan en procesar una secuencia de entrada y generar secuencias de salida a partir de esa entrada. Las tareas más utilizadas son: Traducción automática, Resumen de textos, Respuesta a preguntas o creación de Chatbots, las cuales se resumen en la figura C.12*

Existen dos tipos de modelos pertenecientes a la estructura ***seq2seq*** *que destacan hoy en día:* ***BART*** *y* ***T5****.*

El modelo ***BART (Bidirectional and Auto-Regresive Transformer)*** *[41] con un codificador bidireccional como el modelo* ***BERT*** *y un decodificador autorregresivo como* ***GPT*** *e intenta obtener los mejores resultados de ambos mundos. El codificador empleado tiene como objetivo la eliminación de ruido, mientras que el decodificador intenta reproducir la secuencia original, palabra por palabra, utilizando las palabras anteriores y la salida del codificador. Una ventaja significativa que presenta este modelo es que se puede utilizar con un esquema de ruido arbitrario.*

Por otro lado, aparece también el modelo T5 [9], denominado así porque sus siglas son: Text to Text Transfer Transformer. Es un modelo que todavía se considera estar en pleno desarrollo, aunque a día de hoy ya es capaz de obtener resultados muy buenos. Se caracteriza por ser multitarea, es decir, utiliza el mismo modelo, la misma función

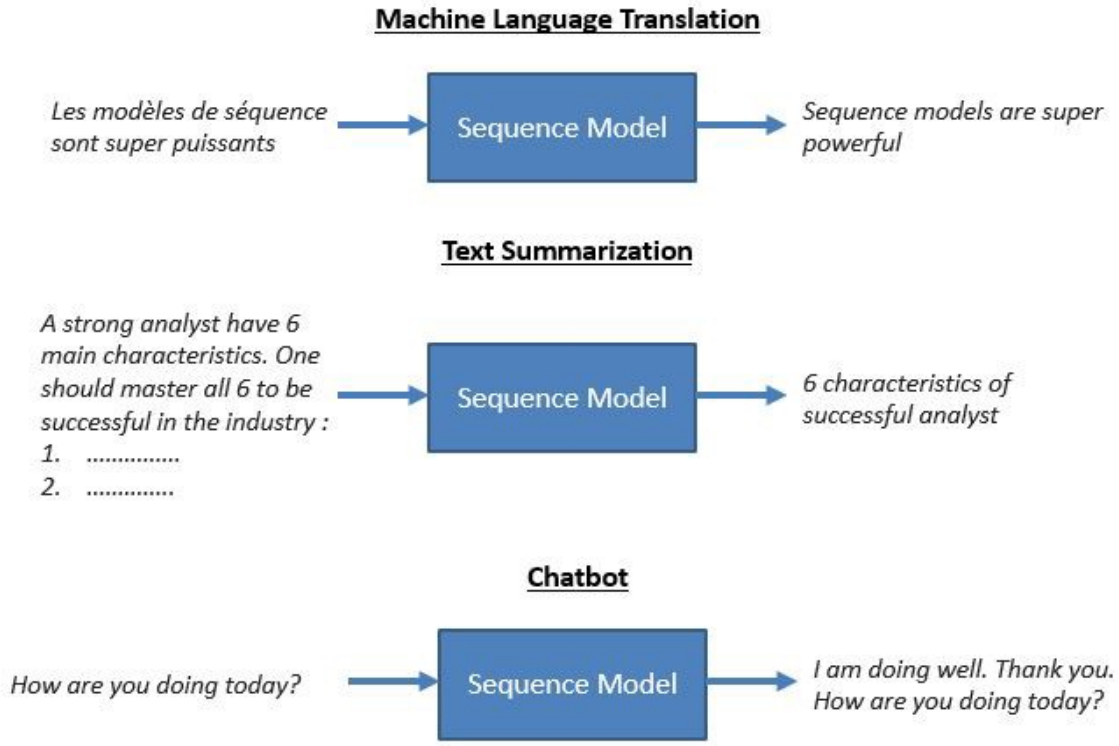

Figura C.12: Tareas principales de los modelos ***seq2seq*** [40]

de pérdida y los mismos hiperparámetros para ejecutar varias tareas. Estas pueden ser: traducción, clasificación de texto, resumidor... El hecho de que el modelo sepa realizar varias tareas introduce una nueva idea, el aprendizaje por transferencia. Si una tarea es rica en datos, esos datos también van a resultar útiles para otro tipo de tareas. Esto ha resultado ser una técnica muy poderosa en el procesamiento del lenguaje natural.