UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Departamento de Sistemas, Informáticos y Computación

Universitat Politécnica de Valéncia

Traducción multilingüe neuronal

Trabajo de fin de Máster

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

AUTOR Jorge Cuevas Muñoz
TUTOR Francisco Casacuberta Nola

Curso 2019/2020

Agradecimientos

Agradezco el apoyo que me ha brindado mi familia, en especial a mi madre, quien me ha apoyado en todas las decisiones que he tomado. A mi tío Iván y mi amigo de infancia Mauricio "Jia" Ramos, quienes me introdujeron en el mundo de la computación.

También agradecer a mi tutor, Dr. Francisco Casacuberta, por el apoyo y la orientación entregada en este estudio. Como también a Álvaro y Miguel, quienes sin conocerme, me entregaron orientación e información más que útil en los momentos más críticos del trabajo.

Finalmente, quisiera agradecer a todas las personas que he conocido durante este tiempo lejos de casa. Partiendo por mis amigos de máster: Janik, Mónica y Pasqual. Mis compañeros de piso, Solange, Adolfo e Isabel, quienes siempre me apoyaron en especial a mi amigo Andrea. A Laetitia y Diego Claure, quienes sin comprender nada de lo que hacía en mi computador, me dieron el ánimo sin siquiera saberlo. Finalmente, a todas aquellas personas que vivieron este proceso conmigo y me alentaron de alguna u otra forma.

Resumen

Los sistemas clásicos de traducción automática están basado en el entrenamiento de pares de lenguas bilingües. Un modelo es diseñado y entrenado para traducir de una lengua fuente a una lengua destino. En los últimos años, se han ido desarrollando nuevas aproximaciones que intentan mejorar la calidad de la traducción con la ayuda de dos o más lenguas en un mismo modelo. En este escenario una lengua es traducida a dos o más lenguas, muchas lenguas pueden ser traducidas a una lengua, o muchas lenguas pueden ser traducidas a muchas lenguas. Esta aproximación es prometedora, diversos estudios han demostrado que utilizar modelos multilingües mejora la traducción de una lengua, sin embargo, esta aproximación conlleva una cantidad no menor de retos dependiendo del diseño del modelo de traducción.

En este trabajo se presenta el entrenamiento de dos tipos de aproximaciones dentro de los modelos de traducción multilingüe en NMT-Keras. La traducción multilingüe con parámetros totalmente compartidos y la traducción automática multilingüe con parámetros controladamente compartidos. Ambas aproximaciones utilizan redes neuronales sobre la clásica arquitectura de codificador, decodificador, además de un modelo atencional. Por otro lado, para lograr construir y entrenar un modelo multilingüe con parámetros controladamente compartidos en NMT-Keras, se modificó el toolkit creando y asignando un decodificador por cada lengua objetivo.

Análogamente, con un corpus reducido y por cada aproximación multilingüe y por cada linea base por lengua, se prueban distintas configuraciones de modelos de traducción automática, con el fin de encontrar la mejor configuración, la cual será entrenada con la totalidad de los datos.

Finalmente se exponen los resultados de cada aproximación, incluyendo las lineas bases. Se comparan los resultados entre las aproximaciones expuestas y con trabajos publicados en la literatura.

Abstract

Classic machine translation systems are based on the training of bilingual language pairs. A model is designed and trained to translate from a source language to a target language. In recent years, new approaches have been developed that attempt to improve the quality of translation with the help of two or more languages in the same model. In this scenario one language is translated into two or more languages, many languages can be translated into one language, or many languages can be translated into many languages. This approach is promising, several studies have shown that using multilingual models improves the translation of a language, however, this approach entails a number of challenges depending on the design of the translation model.

In this work, the training of two types of approaches is presented within the multilingual translation models in NMT-Keras. Multilingual translation with fully shared parameters and multilingual machine translation with controlled shared parameters. Both approaches use neural networks on the classic architecture of encoder, decoder, as well as an attentional model. On the other hand, in order to build and train a multilingual model with controlled parameters shared in NMT-Keras, the toolkit was modified by creating and assigning a decoder for each target language.

Similarly, with a reduced corpus and for each multilingual approach and for each baseline per language, different configurations of machine translation models are tested, in order to find the best configuration, which will be trained with all the data.

Finally, the results of each approximation are exposed, including the baselines. The results are compared between the exposed approaches and with works published in the literature.

Resum

Els sistemes clàssics de traducció automàtica estan basat en l'entrenament de parells de llengües bilingües. Un model és dissenyat i entrenat per a traduir d'una llengua font a una llengua destinació. En els últims anys, s'han anat desenvolupant noves aproximacions que intenten millorar la qualitat de la traducció amb l'ajuda de dues o més llengües en un mateix model. En aquest escenari una llengua és traduïda a dues o més llengües, moltes llengües poden ser traduïdes a una llengua, o moltes llengües poden ser traduïdes a moltes llengües. Aquesta aproximació és prometedora, diversos estudis han demostrat que utilitzar models multilingües millora la traducció d'una llengua, no obstant això, aquesta aproximació comporta una quantitat no menor de reptes depenent del disseny del model de traducció.

En aquest treball es presenta l'entrenament de dos tipus d'aproximacions dins dels models de traducció multilingüe en NMT-keras. La traducció multilingüe amb paràmetres totalment compartits i la traducció automàtica multilingüe amb paràmetres controladament compartits. Ambdós aproximacions utilitzen xarxes neuronals sobre la clàssica arquitectura de codificador, descodificador, a més d'un model d'atenció. D'altra banda, per a aconseguir construir i entrenar un model multilingüe amb paràmetres controladament compartits en NMT-Keras, es va modificar el toolkit creant i assignant un descodificador per cada llengua objectiu.

Anàlogament, amb un corpus reduït i per cada aproximació multilingüe i per cada línia base per llengua, es proven diferents configuracions de models de traducció automàtica, amb la finalitat de trobar la millor configuració, la qual serà entrenada amb la totalitat de les dades.

Finalment s'exposen els resultats de cada aproximació, incloent les línies bases. Es comparen els resultats entre les aproximacions exposades i amb treballs publicats en la literatura.

Índice general

1. Traducción Automática **1**
1.1. Traducción Automática Estadística 2
1.2. Traducción Automática Neuronal 3
1.2.1. Redes neuronales recurrentes 3
1.2.2. Codificador-Decodificador 4
1.2.3. Transformer 6

2. Traducción automática multilingüe **9**
2.1. Traducción automática multilingüe 10
2.1.1. Modelos con parámetros totalmente compartidos 12
2.1.2. Modelos con parámetros parcialmente compartidos 12
2.2. Estado del Arte 14

3. NMT-Keras **17**
3.1. TAM con parámetros compartidos 18
3.2. TAM con parámetros controladamente compartidos 20

4. Marco experimental y Resultados **23**
4.1. Marco experimental 23
4.2. Resultados 29

5. Trabajos Futuros y Conclusiones **35**
5.1. Trabajos Futuros 35
5.2. Conclusiones 36

Bibliografía **37**

Índice de figuras

1.1. Red neuronal recurrente 4
1.2. Configuración Codificador-Decodificador. 5
1.3. Arquitectura de un transformer. 8

2.1. MNMT categorizado en base a casos de usos, problemas centrales y desafíos a resolver [14]. 9
2.2. Paradigma de la traducción automática multilingüe 12
2.3. Configuración de un lenguaje a muchos con un decodificador por lengua objetivo. 13
2.4. Configuración de un lenguaje a muchos con decodificadores separados por lengua y parámetros compartidos. 13

3.1. Arquitectura de un traductor automático en NMT-Keras basado en en el modelo atencional de Bahdanau [9], el cual está compuesto por una capa de embedding, seguido de una red neuronal recurrente bidireccional la cual representa al codificador, el cual se conecta directamente al mecanismo atencional, creando una entrada para el decodificador con la representación dada por el codificador en forma de anotaciones. 18
3.2. Configuración resultante en NMT-Keras para que permita la traducción de una lengua a dos lenguas. 21
3.3. Entrenamiento del modelo propuesto en NMT-Keras para el caso de la traducción del inglés al español-francés. 22

4.1. BLEU por epoch resultante para la linea base. Una epoch equivale a la presentación completa de las muestras de entrenamiento por cada lengua. 26
4.2. BLEU por epoch resultante para el modelo con parámetros totalmente compartidos. Una epoch equivale a la presentación completa de las muestras de entrenamiento para ambas lenguas unidas. 28
4.3. BLEU por epoch resultante para el modelo con parámetros controladamente compartidos. Una epoch equivale a la presentación completa de las muestras de entrenamiento por cada submodelo. 29

Índice de tablas

4.1. Tamaño del corpus de entrenamiento para todas las lenguas de destino. 24
4.2. Tamaño del corpus de validación para todas las lenguas de destino. . 24
4.3. Tamaño del corpus de prueba para todas las lenguas de destino. . . . 24
4.4. Tamaño del corpus reducido para la linea base. 25
4.5. Mejores configuraciones encontradas. 26
4.6. Linea base obtenida sobre el conjunto de prueba, para el español y francés utilizando un corpus reducido. 26
4.7. Tamaño del corpus de entrenamiento para TAM con parámetros totalmente compartidos. 27
4.8. BLEU y TER obtenido sobre el conjunto de prueba, con el corpus reducido para el español, francés y ambos corpus concatenados. . . . 27
4.9. BLEU y TER obtenido sobre el conjunto de prueba, con el corpus reducido para el español y el francés con un decodificador para cada lengua. 28
4.10. BLEU y TER obtenido sobre el conjunto de prueba, con el corpus completo para el español, francés y alemán. 29
4.11. BLEU y TER obtenido sobre el conjunto de prueba, con la totalidad del corpus para el español francés y alemán. El BLEU y TER de los lenguajes de destino Español:Francés, Español:Alemán y Francés:Alemán, son calculados sobre el corpus concatenado de ambas lenguas. 30
4.12. BLEU y TER obtenido sobre el conjunto de prueba, con la totalidad del corpus para el español, francés y alemán 31

Prefacio

Desde hace muchos años, la traducción de lenguas ha sido un problema en diversos ámbitos para las sociedades. Desde el siglo XVII hasta nuestros días, se han ido buscando y desarrollando teorías, métodos y máquinas de traducción para lograr la comunicación entre sociedades con bases lingüísticas diferentes. Este problema se acentúa a medida que en el mundo en que vivimos se globaliza a un paso raudo. En el pasado la comunicación no era expedita, mucho menos en tiempo real, es por lo anterior que en el mundo moderno y altamente globalizado en el que nos encontramos, estamos obligados a tener el conocimiento o herramientas para poder comunicarnos con nuestros pares nativos y extranjeros de forma inmediata, ya sea, por diversos motivos: trabajo, viajes, estudios, etc.

La traducción automática (TA), ha sido blanco de estudio durante varios años, remontándose a 1933 cuando George Artsrouni (franco-armenio) y Petr Smirnov-Troyanskii (ruso) postularon una de las primeras aproximaciones a la traducción automática, fabricando un glosario automático mecánico en el cual se podía traducir una palabra de un idioma otro. Años más tarde, el estadounidense y criptógrafo Warren Weaver planteó el problema de la traducción de lenguas como un problema criptográfico, de tal modo que, una frase en una lengua A estaba codificada con símbolos extraños en una lengua B. Más tarde el mismo Warren propondría los primeros métodos estadísticos para la traducción automática utilizando ordenadores (los cuales habían sido inventados recientemente), dando los primeros lineamientos para la traducción automática moderna.

Años más tarde, aparecerían los sistemas de traducción basados en reglas, en los cuales, se extraía la información de diccionarios y reglas gramaticales por traductores humanos, para generar las traducciones. El fin de estos sistemas basados en reglas fue en 1989 cuando se presentaron los traductores automáticos basados en corpus.

Los sistemas basados en corpus, utilizan pares de sentencias, en los cuales cada sentencia en la lengua X tiene su propia traducción en la lengua Y. Normalmente a estos corpus se les llama corpus paralelos. Dentro de estos sistemas de traducción se ha estado destacando hasta no hace mucho, los sistemas de traducción automática estadística (TAE).

Los esfuerzos por mejorar la TA han puesto en evidencia que no se ha logrado resolver el problema en su totalidad. Hoy en día se utilizan grandes redes neuronales para realizar las traducciones. Las prestaciones computacionales son extremadamente mejores que las del pasado, permitiendo la utilización de diversas técnicas.

A pesar de todos los avances tecnológicos y los resultados obtenidos con las dis-

tintas aproximaciones durante la historia de la TA, se siguen buscando métodos para mejorar la calidad de los traductores automáticos. Investigadores en universidades y el sector privado, han descubierto y demostrado que la utilización de más de un par de lenguas en un mismo modelo, mejoran la calidad de la traducción.

En el presente trabajo se aborda el problema de la traducción automática multilingüe, desarrollando un modelo capaz de traducir de una lengua a dos lenguas diferentes. El documento está estructurado de la siguiente forma: en el Capítulo 1 se introduce la traducción automática y sus desarrollos hasta el día de hoy. En el Capítulo 2 se desarrolla la traducción automática multilingüe en conjunto con el estado del arte en esta área. En el Capítulo 3 se introduce el toolkit NMT-Keras, el cual es utilizado para entrenar dos tipos de modelos dentro de la traducción automática multilingüe, los cuales son: modelo de traducción automática con parámetros totalmente compartidos y modelo de traducción automática con parámetros controladamente compartidos. En el mismo capítulo se presenta la modificación del toolkit, el cual en un principio no posee la característica de entrenar modelos multilingües. En el Capítulo 4 se presentan los experimentos realizados, los resultados obtenidos y su discusión. Finalmente en el Capítulo 5 se presentan los trabajos futuros y conclusiones del presente estudio.

Capítulo 1

Traducción Automática

La traducción automática (TA) es una rama de la lingüística computacional que aborda el cómo traducir texto o habla de una lengua a otra utilizando herramientas computacionales. El problema de la traducción, se remota a varios siglos atrás, incluso a la época de grandes filósofos como Descartes y Leibniz, los cuales fueron los pioneros en buscar relaciones de palabras entre dos lenguas distintas. Durante años se han ido desarrollando distintos métodos, modelos y teorías de sobre cómo diseñar un buen traductor de lengua, incluso se han postulado teorías de lenguajes universales utilizando principios lógicos y símbolos. Hoy en día, el problema de la traducción automática es la calidad de la traducción, además de la búsqueda de un traductor universal. Lo anterior se puede demostrar debido a que existen traductores automáticos especializados en áreas específicas.

La traducción automática, ha ido tomando importancia con el pasar de los años, la globalización ha puesto en aceleración varios aspectos sociales y tecnológicos, y la traducción no ha estado exento de ello. Prueba de lo anterior son las organizaciones internacionales, las redes sociales y la facilidad que hoy en día tienen las personas de moverse por el mundo. Lo anterior, genera la necesidad de poder comprender las distintas lenguas con las que nos enfrentamos día a día, en viajes, en internet, en conferencias, etc.

Las distintas aproximaciones para enfrentar los problemas de la TA pueden ser divididas basados en los siguientes criterios [1]:

1. Dependiendo del tipo de entrada: texto o habla.
2. Dependiendo del tipo de aplicación en la que se usará las traducciones:
    - Aplicaciones que buscan la traducción en una base de datos.
    - Aplicaciones que producen una traducción estimada y luego se mejoran con posedición con la ayuda del usuario.
    - Aplicaciones que generan traducciones interactivas con la ayuda del usuario.
    - Aplicaciones completamente automatizadas para realizar la traducción (restringido al dominio).

3. Dependiendo de la tecnología que se emplea al traducir:

    - Sistemas basados en reglas
    - Sistemas basados en corpus

La utilización de sistemas basados en reglas fue predominante durante algunos años, sin embargo, este sistema de traducción era muy costoso y complejo, además de no ser escalable y adaptable a los cambios del lenguaje, puesto que, requerían de actualizaciones de las reglas que modelan el lenguaje.

1.1. Traducción Automática Estadística

Los sistemas basados en corpus han dominado el campo de la TA durante los últimos años, destacando en ellos la traducción automática estadística (TAE), la cual fue presentada en los años 50 y retomada en los 90, debido a la mejora de las prestaciones computacionales. La TAE ha demostrado ser el estado del arte en este tópico durante varios años no muy lejanos al actual. La hipótesis principal que propone la TAE es que para cada frase del lenguaje fuente existe una posible traducción en la lengua de destino.

Más formalmente, dado una sentencia x en un lenguaje fuente, el sistema TAE busca una traducción apropiada y en el lenguaje destino. Lo anterior puede ser modelado de la siguiente forma [2]:

$$\hat{y} = arg \max_{y} Pr(y|x) \tag{1.1}$$

la ecuación anterior entrega la frase de destino y que maximice la probabilidad de que y sea una traducción de la frase x del lenguaje fuente dado.

Por otra parte, los enfoques más recientes de TAE se basan en el modelo log-linear para $Pr(y|x)$ el cual es modelado de la siguiente forma [3]:

$$\hat{y} = arg \max_{y} \left\{ \sum_{n=1}^{N} \lambda_n \cdot log(f_n(y,x)) \right\} \tag{1.2}$$

donde $log(f_n(y,x))$ puede ser una característica importante para el modelo de traducción, N es el número de características y λ_n son los pesos de la combinación log-linear.

Análogamente, TAE se enfrenta a dos desafíos, los cuales son:

- Aprendizaje de los modelos que generan las características.
- Buscar un método eficiente y efectivo para la búsqueda global.

Por otra parte, existen los métodos basados en redes neuronales, los cuales se han utilizado en este trabajo y se les dará mayor énfasis en la siguiente sección.

1.2. Traducción Automática Neuronal

La traducción automática neuronal TAN (o en inglés NMT de neuronal machine translation) se ha convertido en una aproximación importante en la traducción automática, no solo en la investigación académica sino que también en las empresas privadas que la utilizan para uso comercial [4]. TAN ha demostrado tener un mejor rendimiento en muchos casos frente a la TAE clásica, incluso alcanzando el estado del arte en diferentes pares de lenguas [5], gracias a la utilización de redes neuronales. La idea principal de la TAN se representa con la siguiente ecuación:

$$\hat{s} = arg\,\underset{s}{max} \prod_{i=1}^{|s|} p_{r_\theta}(s_i|s_1^{i-1}, c(e)), \tag{1.3}$$

donde s_i es la palabra traducida actual, la cual se genera a partir de las palabras s_1^{s-1} que fueron traducidas con anterioridad con un tipo de representación denotado por la función c de la sentencia de origen e, y de los parámetros del modelo θ.

Aunque el uso de redes neuronales para la traducción automática se propuso hace ya varios años [24], hoy en día se han convertido en la herramienta más utilizada para la TA, incluso se han desarrollado bastantes aproximaciones para la TAN, comúnmente siguiendo el paradigma de codificador-decodificador como base, además de la utilización de grandes redes neuronales.

1.2.1. Redes neuronales recurrentes

Una red neuronal recurrente (RNN) es un tipo de red neuronal que contiene un estado oculto h y una salida opcional y que opera en una secuencia de tamaño variable $x = (x_1, ..., x_T)$. En cada tiempo t, el estado oculto $h_{(t)}$ es actualizado de la siguiente manera:

$$h_t = f_h(x_t, h_{t-1}) \tag{1.4}$$

$$y_t = f_o(h_t) \tag{1.5}$$

donde h_t es el estado oculto de la red neuronal en el tiempo t. La RNN puede aprender la distribución de probabilidad sobre una secuencia siendo entrenada para predecir el símbolo siguiente de una secuencia. La salida en cada tiempo t es la distribución condicional $p(X_t|x_{t-1}, ...x_1)$. Diferentes elecciones de h, f_h y f_o generaron nuevas arquitecturas de redes neuronales como las redes de Jordan [30] o las redes de Elman [31]. En la Figura 1.1 se puede observar la arquitectura de Elman. En este caso las ecuaciones 1.4 y 1.5 son reescritas de la siguiente forma:

$$h_t = f_h(x_t, h_{t-1}) = \phi(W^T h_{t-1} + U^T x_t) \tag{1.6}$$

$$y_t = f_o(h_t) = \sigma(V^T h_t) \tag{1.7}$$

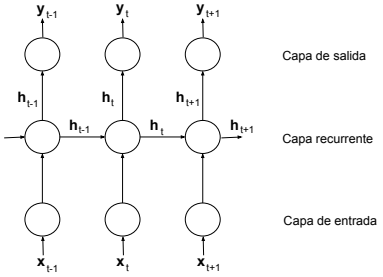

Figura 1.1: Red neuronal recurrente

donde W, U y V son los pesos de las matrices de la capa recurrente, la capa de entrada, y la capa de salida respectivamente. Por otra parte, ϕ es una función de activación no lineal (usualmente sigmoidea o tangente hiperbólica). Finalmente σ es comúnmente una función softmax la cual se define de la siguiente forma:

$$\sigma(z_k) = \frac{exp(z_k)}{\sum_{k'=1}^{K} exp(z_{k'})} \tag{1.8}$$

donde z_k es la k-ésima unidad de salida. Cada unidad de salida representa una palabra del diccionario de palabras, por lo que el tamaño de la capa de salida es equivalente al tamaño del vocabulario.

1.2.2. Codificador-Decodificador

Un codificador-decodificador (Figura 1.2) en una red neuronal codifica una secuencia de longitud variable en una representación vectorial de longitud fija. Por otra parte, decodifica el vector de longitud fija en una secuencia de longitud variable. Los componentes del codificador-decodificador se entrenan con el fin de maximizar la probabilidad logarítmica condicional dado un conjunto de pares de entrenamiento $X = \{(x_1, y_1), ..., (x_N, y_N)\}$. Lo anterior se puede modelar de la siguiente forma:

$$arg\,\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} log\rho_\theta(y_n|x_n) \tag{1.9}$$

donde $x_n = x_1, ..., x_J$ es una sentencia de entrada de tamaño J, $y_n = y_1, ..., y_I$ es una sentencia de salida de tamaño I y finalmente θ es el conjunto de parámetros.

La entrada de estas redes es el lenguaje fuente, y su output es el lenguaje destino (u objetivo) el cual posee la traducción real.

Codificador

El codificador normalmente está compuesto por una red neuronal recurrente con capas ocultas del tipo "Long Short Term Memory" (LSTM) [34] o "Gated Recurrent Unit" (GRU) [35]. El codificador proyecta una representación de su entrada en un

vector de tamaño fijo, mientras que el decodificador es alimentado con esta representación para producir la traducción de la sentencia en el lenguaje objetivo [6], como se muestra en la Figura 1.2:

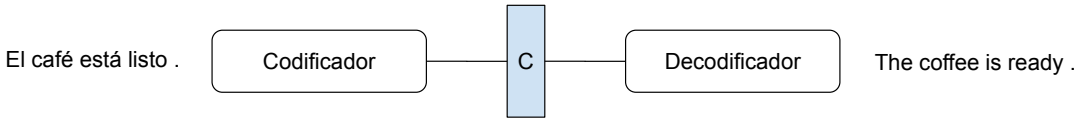

Figura 1.2: Configuración Codificador-Decodificador.

Una diferencia que caracteriza a la traducción automática neuronal es la representación de las palabras y frases, las cuales son proyectadas en vectores numéricos [8].

El codificador formalmente se modela de la siguiente forma [6, 7]:

$$h_j = f(x_j, h_{j-1}) \tag{1.10}$$

y

$$c = q(h_1, ..., h_J) \tag{1.11}$$

donde x_j es una secuencia de entrada al modelo, $h_j \in \mathbb{R}^n$ es el estado oculto en el tiempo j, y c es un vector generado en base a la secuencia de estados ocultos, mientras que f y q son funciones no lineales.

Decodificador

El decodificador es entrenado para generar la sentencia de salida $y_1^I = y_1, ..., y_I$ dado las palabras predichas anteriormente y el vector de contexto c. Desde una perspectiva probabilística el decodificador se define como:

$$p(y) = \prod_{i=1}^{I} p(y_i|\{y_1, ..., y_{i-1}\}, c) \tag{1.12}$$

donde $y = (y_1, \cdots, y_I)$. Utilizando una red neuronal recurrente (RNN) como es en este caso, la probabilidad se modela de la siguiente forma:

$$p(y_i|\{y_1, ..., y_{i-1}\}, c) = g(y_{i-1}, s_i, c), \tag{1.13}$$

donde g es una función no lineal y s_i es el estado oculto de la RNN.

Modelo atencional

El uso de un tamaño fijo de vectores en el modelo de codificador-decodificador produce un cuello de botella, dado que el vector al ser de tamaño fijo no puede capturar el contexto completo de la sentencia cuando esta es de un tamaño extenso [10]. Sin embargo, este problema se soluciona creando un vector de contexto variable, el

cual captura el contexto y las relaciones de las sentencias a traducir [9]. Formalmente esta aproximación se modela de la siguiente forma:

$$p(y_i|\{y_i, ..., y_{i-1}\}, c_i) = g(y_{i-1}, s_i, c_i), \tag{1.14}$$

donde g es una función no lineal, s_i es el estado oculto de la RNN y el vector c_i es la suma ponderada de la secuencia de anotaciones salientes del codificador, la cual se representa matemáticamente como:

$$c_i = \sum_{j=1}^{J} \alpha_{ij} h_j \tag{1.15}$$

donde α_{ij} es estimado por una función softmax para cada h_j. Esta estimación se realiza con la siguiente ecuación:

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{J} exp(e_{ik})} \tag{1.16}$$

donde $e_{ij} = a(s_{i-1}, h_j)$ es una puntuación dada por un modelo de alineamiento, el cual mide que tan bien las entradas de posición j coinciden con las posiciones i de las salidas.

Este modelo de alineamiento recibe el estado previo del decodificador s_{i-1}, y la anotación de la sentencia fuente h_j. Formalmente este modelo se describe de la siguiente forma:

$$A(s_{i-1}, h_j) = v_a^\top tanh(W_a S_{i-1} + U_a hj) \tag{1.17}$$

donde v_a, W_a y U_a, son los pesos entrenables del modelo atencional. Esto permite obtener una representación de la sentencia de entrada basada en el entrenamiento de estos pesos.

Existen otras aproximaciones en la literatura para realizar el cálculo del modelo atencional, como el producto punto atencional [33] el cual formalmente se escribe de la siguiente forma:

$$A(s_{i-1}, h_j) = s_{i-1}^\top h_j \tag{1.18}$$

Esta aproximación posee una ventaja debido a que no posee parámetros como en la ecuación 1.17. Esta ventaja radica en el simple producto punto entre los vectores de proyección de consulta (en inglés "query") y llave (en inglés "key").

1.2.3. Transformer

El transformer [32] es una arquitectura que se ha introducido recientemente, el cual reemplaza las capas recurrentes por una nueva capa de autoatención, además de varios cambios en la arquitectura tradicional de codificador-decodificador.

El codificador y decodificador están compuesto por la unión de varias capas de autoatención, las cuales forman bloques en la parte superior de cada componente.

El codificador está compuesto por la unión de 6 capas idénticas. Cada capa tiene un conjunto de capas secundarias con distintas funcionalidades. La primera capa

contiene un mecanismo multicabezal de autoatención, mientras que la segunda es una capa totalmente conectada. Por otra parte, existe una conexión residual entre cada una de las dos capas secundarias, seguidas de una capa de normalización.

El decodificador está compuesto por la unión de 6 capas al igual que el codificador, con la diferencia de que se introduce una tercera capa secundaria la cual utiliza un muticabezal atencional sobre la salida de las capas del decodificador.

Esta arquitectura contiene múltiples productos puntos escalares atencionales usando un multicabezal atencional. El producto punto fue introducido en la ecuación 1.18. Este producto punto atencional tiene una entrada de consulta (la cual usualmente es el estado oculto del decodificador) y llaves de tamaño d_k (estados ocultos del codificador), y valores de dimensión d_v (pesos normalizados), los cuales representan cuanta atención posee una llave. El producto es calculado por la consulta para todas las llaves, dividido por cada $\sqrt{d_k}$ y aplicando una función de activación softmax para obtener los pesos de cada valor.

En la práctica, el cálculo se realiza sobre un conjunto de consultas simultáneamente encapsuladas en una matriz Q. Por otra parte, las llaves y los valores son encapsulados en las matrices K y V respectivamente. Formalmente esto se presenta de la siguiente forma:

$$A(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.19}$$

Con el cálculo anterior, el mecanismo de atención solo nos entrega una respuesta para cada consulta. El multicabezal atencional permite realizar el cálculo anterior a un nivel de múltiples consultas y llaves. Lo anterior desencadena en varios cálculos atencionales en paralelo combinando el resultado para obtener un vector de contexto. El multicabezal atencional se calcula se la siguiente forma:

$$Multicabezal(Q, K, V) = Concat(cabezal_1, ..., cabezal_h)W^O, \tag{1.20}$$

donde

$$cabezal_i = A(QW_i^Q, KW_i^K, VW_i^V) \tag{1.21}$$

Las proyecciones son matrices de parámetros $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ y $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, los cuales son aprendidos durante la etapa de entrenamiento.

En la Figura 1.3 se muestra la arquitectura de un transformer con 6 codificadores y 6 decodificadores.

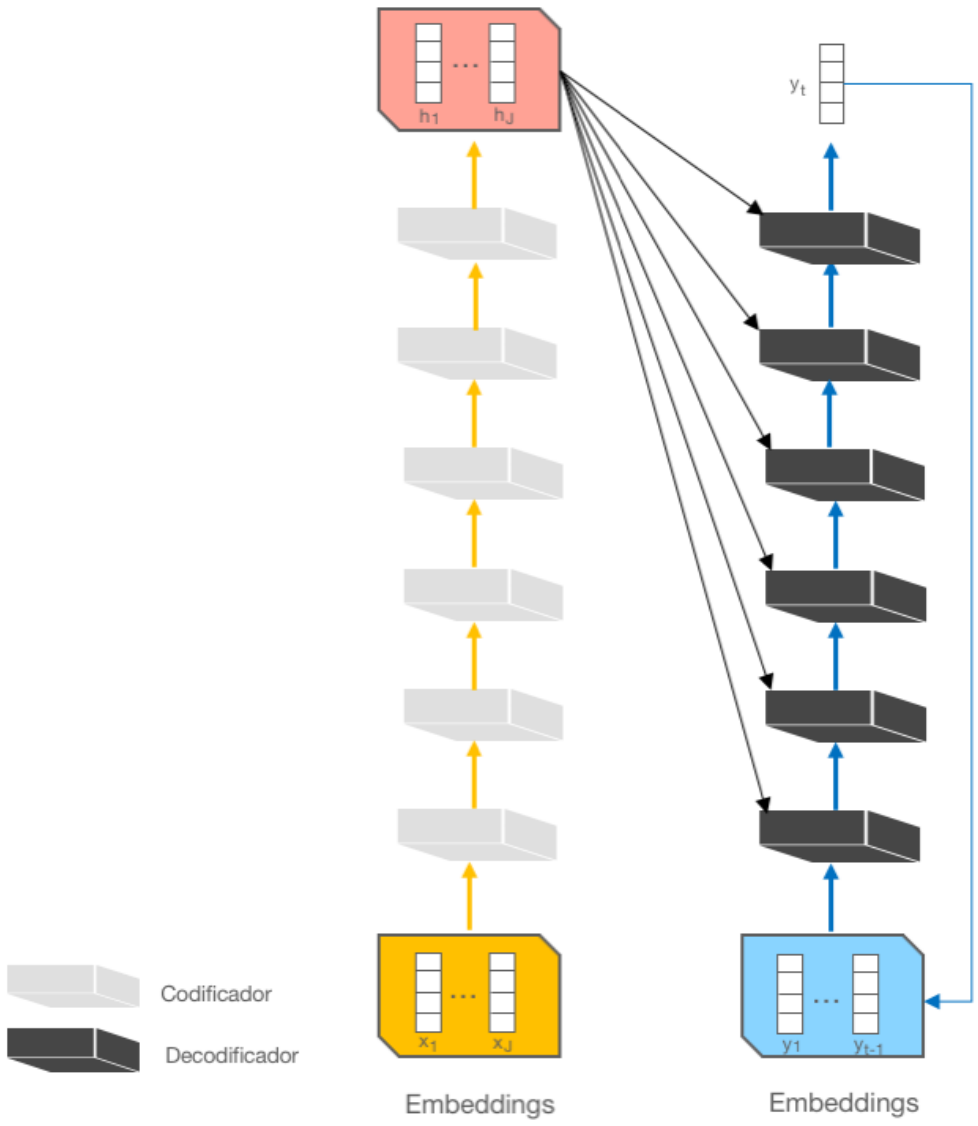

Figura 1.3: Arquitectura de un transformer.

Capítulo 2

Traducción automática multilingüe

Hasta hace poco, la idea inicial de estudiar la traducción automática fue basada en la traducción de un lenguaje a otro (Sección 1). En los últimos años investigadores han descubierto que la utilización de múltiples lenguajes en los sistemas de TA mejoran la calidad de la traducción [11, 12, 13]. A este nuevo sistema de traducción se le denomina en inglés como "multilingual neuronal machine translation" (MNMT). A diferencia del sistema de pares de lenguajes en TAN, MNMT procesa múltiples pares de lenguajes en un mismo modelo.

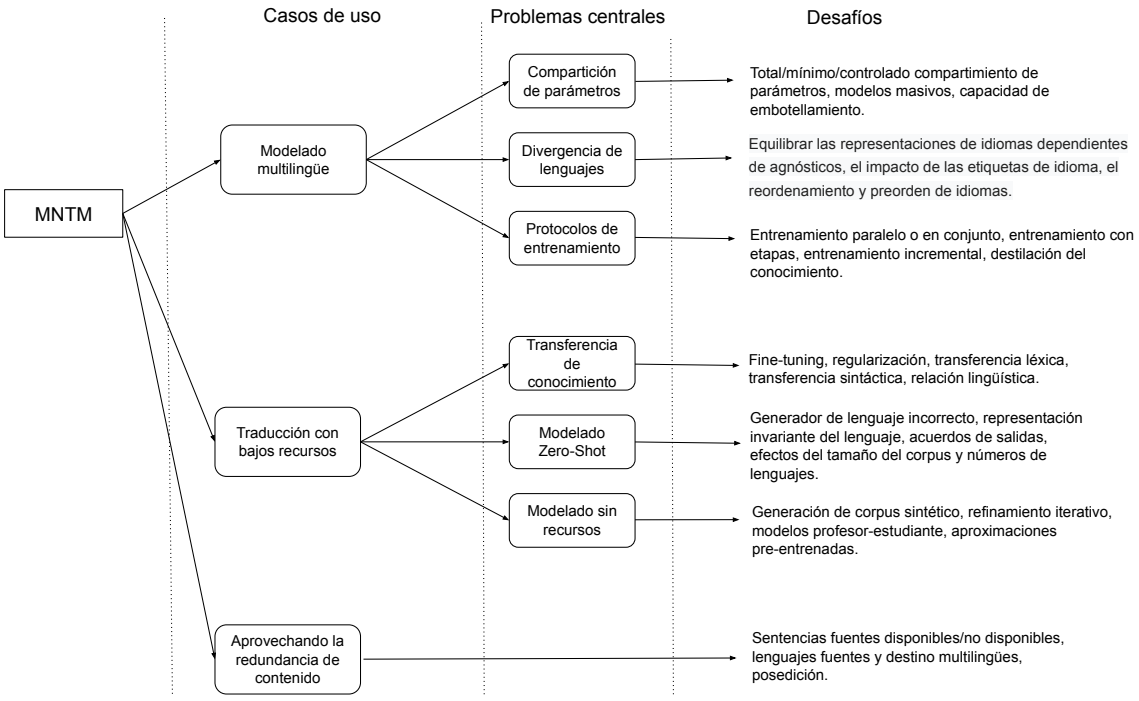

Figura 2.1: MNMT categorizado en base a casos de usos, problemas centrales y desafíos a resolver [14].

El objetivo principal de MNMT es crear un modelo de traducción capaz de utilizar todos los recursos disponibles en las distintas lenguas participantes del proceso.

Los sistemas MNMT están tomando ventaja sobre los de NMT, puesto que, MNMT ha demostrado que ayuda al lenguaje con recursos bajos de traducción a adquirir conocimiento de los otros lenguajes [15]. Por otra parte, MNMT permite que la capacidad de generalización de los modelos aumente dado la exposición de diversos lenguajes en un mismo sistema, aportando conocimiento (transferencia de conocimiento [16]) en las decisiones de desambiguación mejorando la calidad de la traducción. Por otra parte, se han presentado estudios en los cuales se desmuestra que MNMT también aporta a la calidad de la traducción cuando los modelos son entrenados con lenguajes de recursos altos [17]. Comúnmente se pensaría que al añadir más lenguajes a un modelo NMT los parámetros crecerían considerablemente. Lo anterior no es correcto en su totalidad, debido a que un sistema NMT puede ser convertido en un MNMT fácilmente (como se exhibe en la Sección 3.1), manteniendo un modelo compacto.

Como se muestra en la Figura 2.1, existen diversos escenarios que se han estado estudiando durante los últimos años. Dentro de ellos los más explorados en la literatura han sido [14]:

- Traducción multilingüe: el objetivo principal es construir un modelo único que acepte varias lenguas, tales como, de una a varias lenguas, de varias lenguas a una, y de varias lenguas a varias lenguas.

- Traducción de bajos recursos: el objetivo principal es mejorar la traducción de aquellos lenguajes con bajos corpus paralelos o sin ellos. Existen dos tipos en éste tópico:

  a) Pares de lenguajes de altos recursos.

  b) No existe un corpus paralelo directo para la lengua de bajos recursos, pero las lenguas comparten un corpus paralelo con uno o más idiomas pivote.

- Traducción de múltiple recursos: la redundancia existe en el lenguaje fuente, al haber sido traducido en múltiples lenguajes [15]. Esto podría ayudar a mejorar la traducción al aportan en la desambiguación al momento de la traducción al nuevo lenguaje.

En la siguiente sección se extenderá el concepto multilingüe en profundidad.

2.1. Traducción automática multilingüe

Como se dijo anteriormente, el objetivo de MNMT es encapsular un modelo que sea capaz de procesar múltiples lenguajes. Formalmente existen l pares de lenguas $(src_l, tgt_l) \in L(l = 1\,a\,L)$, donde $L \subset S \times T$, y S,T son los conjuntos de datos X del lenguaje fuente e Y del lenguaje objetivo respectivamente. Expuesto lo anterior, es evidente que es necesario tener un corpus para cada par de lenguas, tanto para el lenguaje fuente como para el lenguaje objetivo.

Dentro de la traducción automática multilingüe se pueden distinguir tres clases de modelos, los cuales son agrupados dependiendo del nivel de compartición de parámetros entre lenguas. Las clases y algunas de sus características son [14]:

1. Parámetros mínimamente compartidos: sistemas que encapsulan la mayoría de los parámetros por lengua. Solo algunos de los parámetros son compartidos. Dentro de esta clase predominan una serie de componentes que se listan a continuación:
   - Codificadores y decodificadores separados.
   - Capa atencional compartida.
   - Modelos de gran volumen.
   - Descomponible - separable.
   - Sin cuello de botella.
   - Menos populares.
2. Parámetros parcialmente compartidos: sistemas que comparten controladamente los parámetros. Normalmente estos modelos son complejos. Los componente que destacan en esta clase son:
   - Parámetros controlados.
   - Modelos complejos.
   - Creación contextual de los parámetros del modelo.
   - Posible arquitectura de modelos basada en datos.
   - Están ganando popularidad
3. Parámetros totalmente compartidos: sistemas que comparten todos los parámetros entre todos los lenguajes que se procesan en el modelo. Los componentes predominantes en esta clase son:
   - Codificador, decodificador y capa atencional compartida.
   - Cadena de texto que indican el lenguaje objetivo (etiqueta).
   - Modelos ligeros y compactos.
   - No descomponible o separable.
   - Cuello de botella.
   - Más popular.

Los modelos de traducción automática multilingüe permiten la compartición de conocimiento entre lenguas, ayudan a la contextualización y a la desambiguación. Por otra parte, es posible comprender la relación de los lenguajes desde una perspectiva estadística y lingüistica [18]. En el presente trabajo se presenta un modelo con parámetros totalmente compartidos y un modelo con parámetros compartidos en forma parcial.

En la siguiente sección se estudiarán más en profundidad estos modelos.

2.1.1. Modelos con parámetros totalmente compartidos

En esta aproximación un único modelo es entrenado con múltiples lenguajes, por lo tanto, todos los parámetros del modelo son compartidos. Para realizar esto no es necesario realizar modificaciones en el sistema clásico de TAN (embedding-codificador-Atención-decodificador), debido a que la característica principal de esta aproximación se basa en una etiqueta o token [17], que indica el lenguaje objetivo en el lenguaje fuente, como se muestra en la Figura 2.2:

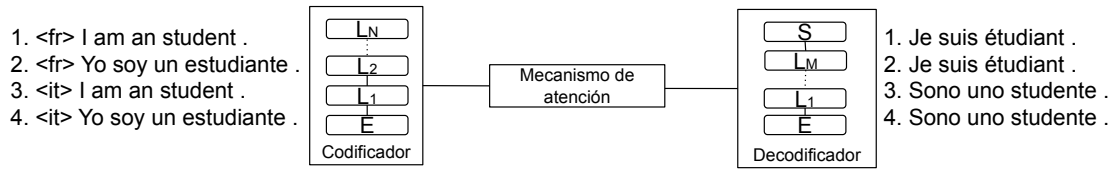

Figura 2.2: Paradigma de la traducción automática multilingüe

En el ejemplo de la Figura 2.2 se define un único codificador, los embeddings (E) y un decodificador compartido por ambos lenguajes. El codificador podría poseer N capas ocultas, de L^1 a L^N, además de una entrada única X la cual contiene los lenguajes fuentes concatenados. Por otra parte el decodificador podría poseer M capas, de L^1 a L^M y una salida única Y que puede ser uno de los lenguajes objetivos. La etiqueta añadida a cada sentencia del lenguaje fuente es aprendida por la red neuronal y es capaz de distinguir el lenguaje objetivo.

Normalmente en esta aproximación se utiliza un vocabulario único a lo largo de cada lenguaje utilizado en el modelo, puesto que, la entrada al modelo son secuencias de distintas lenguas desde un mismo origen, como se muestra en la Figura 2.2. En la práctica, lo anterior se traduce en la construcción de un único vocabulario para distintos lenguajes, debido a la concatenación de los corpus en la entrada y la salida. Usualmente este vocabulario es generado usando técnicas de niveles de sub-palabras como byte-pair encoding (BPE) [19], word-piece model (WPM) [20] o sentence-piece model (SPM) [21].

Por otra parte, el entrenamiento se realiza de igual forma que un sistema TAN, en el cual se minimiza la entropía cruzada como función de pérdida entre la sentencia estimada y la sentencia real. Análogamente, se postuló que este tipo de aproximación es beneficioso para aquellos lenguajes que tienen cierto grado de similitud, o que están relacionados de alguna forma a nivel léxico y sintáctico [22].

2.1.2. Modelos con parámetros parcialmente compartidos

En este tipo de aproximación se decide qué parámetros se compartirán a lo largo del modelo. Se estima que los parámetros a compartir dependen del origen de cada lenguaje y sus similitudes, como también la divergencia de estos [22]. La literatura recomienda poner mayor atención al decodificador, puesto que, este se encarga de hacer la traducción a la lengua objetivo, mientras que el codificador tiene un trabajo más simple en este tipo de tareas.

Análogamente, se estima que al compartir el codificador a lo largo de diferentes lenguajes se reduce la cantidad de parámetros del modelo.

Se han presentado diversos modelos los cuales se diferencian dependiendo de los parámetros que comparten.

En Dong et al., 2015 [23] se presenta la configuración de un codificador compartido y un decodificador por cada lengua, como se muestra en la Figura 2.3. Esta arquitectura no es trivial de construir y de entrenar. Por otra parte, ha ido ganando popularidad en este tipo de tareas, debido a que el decodificador tiene la oportunidad de ser entrenado con un solo lenguaje. Si bien esta arquitectura se está popularizando por su buen desempeño, los parámetros crecen a medida que se integran lenguajes objetivos, como también incrementa la utilización de memoria y el tiempo de entrenamiento.

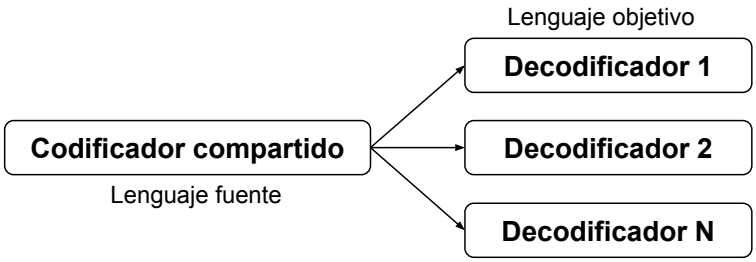

Figura 2.3: Configuración de un lenguaje a muchos con un decodificador por lengua objetivo.

Otra configuración posible para esta tarea, es utilizar parámetros compartidos entre decodificadores [22]. La idea principal de esta aproximación es tener un decodificador por lengua, los cuales comparten algunos componentes de cada decodificador, como se muestra en la Figura 2.4. Este modelo híbrido propone compartir la mayor cantidad de parámetros entre los decodificadores, para aprovechar los beneficios de compartir información entre lenguas objetivo sin perder la particularidad de decodificar cada lenguaje por separado.

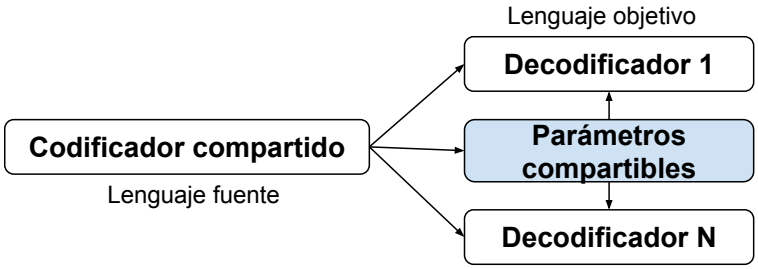

Figura 2.4: Configuración de un lenguaje a muchos con decodificadores separados por lengua y parámetros compartidos.

Este tipo de aproximación es altamente compleja, dado que, se debe mantener un equilibrio entre los parámetros compartidos y la exactitud del modelo, en vista de crear un único sistema compacto.

2.2. Estado del Arte

La traducción automática multilingüe es una nueva aproximación en el área de la traducción automática, poco explorada, la cual puede utilizar múltiples lenguajes fuentes y destinos, o distintas configuraciones entre ellas, como un lenguaje fuente a muchos como en el caso de este trabajo.

Varios modelos han sido presentados en la literatura, debido a los buenos resultados que están demostrando estas aproximaciones, incluso alcanzando el estado del arte en la traducción automática, tanto en la literatura como en la industria. Recientes trabajos han demostrado los beneficios y mejoras que están aportando los modelos multilingües.

En Yiren Wang at al. [40], se propone un aprendizaje multitarea, utilizando un corpus de entrenamiento de bitext y dos modelos de lenguaje de eliminación de ruido en corpus monolingües (lenguaje enmascarado y codificación automática), con el fin de mejorar la traducción. En este trabajo se realiza un estudio extenso sobre TAM con 10 pares de lenguas de la WMT. En este trabajo se muestra que se mejora eficazmente la calidad de la traducción para lenguas de recursos altos y bajos, logrando resultados significativamente mejores que los modelos bilingües individuales. Los autores utilizan la arquitectura de transformadores para todos los experimentos, los cuales están conformados por muchos lenguajes a inglés, ingles a muchos lenguajes, y muchos lenguajes a muchos lenguajes. Los autores añadieron una etiqueta a cada sentencia de entrada la cual representa el lenguaje objetivo de cada sentencia [17]. El trabajo demuestra que el aprendizaje de un modelo multilingüe combinado con aprendizaje multitarea ayuda a mejorar la calidad de la traducción en lenguajes de altos y bajos recursos.

Zehui Lin at al. [36] investigan la pregunta "¿es posible construir un único modelo de traducción automática universal que sirva como semilla común para obtener modelos derivados y mejorados en pares de idiomas arbitrarios?". Los autores proponen mRASP un método de preentrenamiento, el cual propone un enfoque para entrenar un modelo de TAM universal. mRASP utiliza una nueva técnica de sustitución alineada aleatoria. Su funcionamiento se basa en el acercamiento de palabras y frases con significados similares en lenguajes parecidos en el espacio de representación. En el trabajo se entrena el modelo mRASP con 32 pares de lenguas conjuntamente sobre una arquitectura basada en el transformador. Los autores modifican la arquitectura clásica del transformador reemplazando la función de activación ReLU por una GeLU [37] en la red neuronal. Para realizar la detección del lenguaje objetivo, los autores añaden la etiqueta del lenguaje objetivo a las sentencias de entrada [17]. Los experimentos realizados en este trabajo demuestran que el modelo propuesto mejora la calidad de la traducción. Paralelamente, se demuestra también que al utilizar varios lenguajes de bajos recursos en el modelo puede mejorar la calidad de traducción de los lenguajes de altos recursos.

En Gao at at. [39] se presenta DecSDE , un embedding eficiente el cual está basado en n-gramas de caracteres diseñados para el decodificador. El objetivo de este trabajo es mejorar la traducción automática multilingüe para los lenguajes de bajos recursos. Los autores diseñan un método para construir un embedding para el lenguaje objetivo en la traducción automática multilingüe. DecSDE está basado en "Soft Decoupled Encoding" (SDE) [40], pero adaptado al decodificador. La arquitectura utilizada en este trabajo está basada en el transformador con 6 capas de codificadores y decodificadores. El trabajo presenta mejoras en la calidad de la traducción en los lenguajes de bajos recursos al mirar a nivel de caracteres en el vocabulario del lenguaje objetivo cuando se crean las subpalabras y palabras en los embeddings.

Capítulo 3

NMT-Keras

NMT-Keras es un toolkit desarrollado por la Universidad Politécnica de Valencia (UPV) basado en la biblioteca para aprendizaje profundo Keras [26] la cual trabaja sobre la biblioteca de Tensorflow. Posee una serie de herramientas para entrenar modelos de aprendizaje profundo. Además se especializa en tareas de traducción automática, tales como, protocolos de traducción interactiva, predictiva y adaptación del sistema de traducción a través del aprendizaje continuo. NMT-Keras posee dos modelos de traducción automática: Attention RNN (Figura 3.1) [9] y Transformer [27]. Estos pueden ser instanciados rápidamente sin intervención a nivel de código, además de ser fácilmente modificados desde el configurador indicando los hiperparámetros para el modelo, como también para el entrenamiento.

Esta herramienta soluciona el problema de instanciar tareas secuenciales aplicadas a entradas y salidas de texto, de una manera eficaz e intuitiva.

NMT-Keras está compuesto por tres grandes componentes, los cuales se listan a continuación:

- Keras: API diseñada para el fácil uso de cualquier persona con conocimientos de programación. Keras sigue las mejores prácticas para reducir la carga cognitiva, posee una API consistente y simple de utilizar, además de ser intuitiva. Por otra parte, minimiza el número de acciones del usuario para la creación de modelos tradicionales o no complejos.

- Multimodal Keras Wrapper: permite manejar el entrenamiento y aplicación de modelos complejos en Keras. Posee gestión de datos, el cual permite cargar y modificar los datos, como también la instanciación de modelos y llamadas de vuelta (callbacks) durante el entrenamiento.

- NMT-Keras: biblioteca la cual utiliza Keras y Multimedia Keras Wrapper para construir el proceso de traducción automática desde el preprocesamiento hasta la predicción.

En el presente trabajo se adaptará la configuración estándar de NMT-Keras para permitir el entrenamiento de un modelo de traducción automática multilingüe, visto en la sección 2.1. Por otra parte, se modificará la arquitectura base expuesta en la

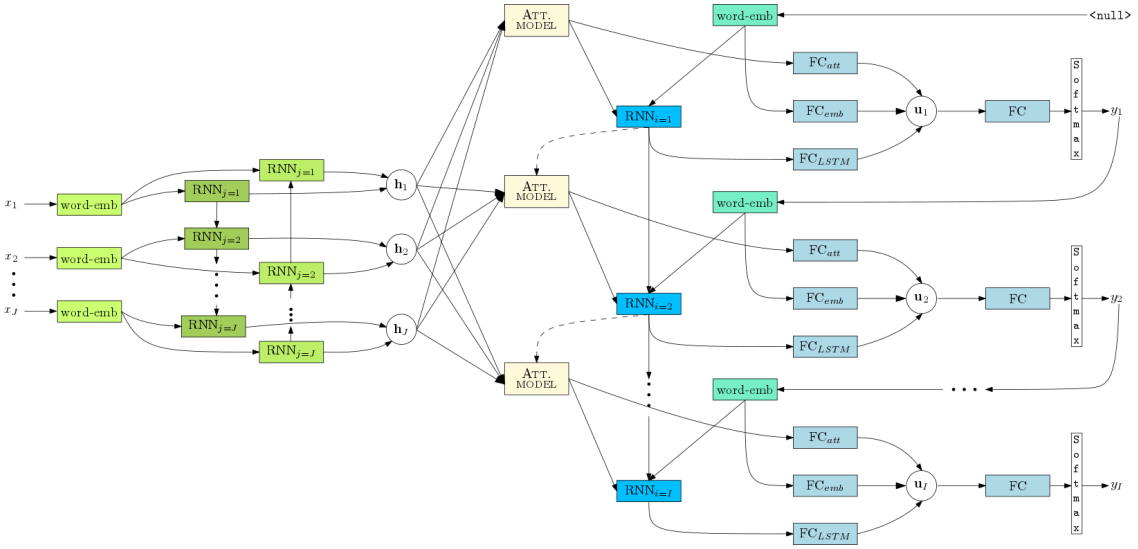

Figura 3.1: Arquitectura de un traductor automático en NMT-Keras basado en en el modelo atencional de Bahdanau [9], el cual está compuesto por una capa de embedding, seguido de una red neuronal recurrente bidireccional la cual representa al codificador, el cual se conecta directamente al mecanismo atencional, creando una entrada para el decodificador con la representación dada por el codificador en forma de anotaciones.

Figura 3.1 con el fin de crear un modelo capaz de entrenar una lengua de origen a dos lenguas objetivos, con decodificadores paralelos, tal como se muestra en la Figura 2.3.

3.1. Traducción automática multilingüe con parámetros completamente compartidos en NMT-Keras

Como se exhibió en la sección 2.1.1, es necesario agregar una etiqueta al principio de cada sentencia de entrada en el lenguaje fuente (ver Figura 2.2). Lo anterior se realizó fuera de NMT-Keras, debido a que el toolkit no está diseñado para realizar este tipo de trabajo, por lo tanto, se abordó en la parte del preproceso y no fue necesaria la modificación del toolkit. Cada sentencia en el lenguaje fuente fue etiquetada con el nombre de su respectivo lenguaje objetivo.

El corpus resultante del lenguaje fuente, para el caso de un modelo de traducción del inglés al español quedó de la siguiente forma:

__trgt__es Resumption of the session
__trgt__es I declare resumed the session of the...
__trgt__es Although, as you will have seen...

y para el inglés al francés:

__trgt__fr Resumption of the session
__trgt__fr I declare resumed the session of the..
__trgt__fr Although, as you will have seen...

Seguido de lo anterior, los corpus fueron concatenados en un único corpus, similar a la siguiente estructura:

__trgt__es Resumption of the session
__trgt__es I declare resumed the session of the...
__trgt__es Although, as you will have seen...
...
__trgt__fr Resumption of the session
__trgt__fr I declare resumed the session of the..
__trgt__fr Although, as you will have seen...

Por otra parte, al crear un nuevo corpus unificado para el lenguaje fuente, fue necesario unir el corpus de los lenguajes objetivos, con el fin de que las sentencias de ingles-español e inglés-francés estén pareadas y se posicionen correctamente, tal como se demuestra en la sección 2.1.1. Por lo anterior, el corpus de lenguajes objetivos quedaría de la siguiente forma:

Reanudación del período de sesiones
Declaro reanudado el período de sesiones
Como todos han podido comprobar
...
Reprise de la session
Je déclare reprise la session...
Comme vous avez pu le constater...

Es importante destacar que no es necesario que ambos lenguajes tengan exactamente las mismas sentencias o la misma cantidad de estas. Como también, destacar que si se quiere añadir un nuevo lenguaje, bastaría con añadir una nueva etiqueta al principio de las sentencias del corpus del lenguaje fuente, indicando el nombre del lenguaje objetivo.

Preproceso y Entrenamiento

Como se mencionó en el apartado anterior, los corpus fueron concatenados. Sin embargo, cada par de corpus fue preprocesado con la técnica de BPE por separado antes de ser concatenados. Esto permite la reducción del vocabulario, el tratamiento de las palabras desconocidas y además una aceleración en el entrenamiento.

El entrenamiento de este tipo de aproximación es bastante sencillo e intuitivo, debido a que, al poseer una lengua de origen única y dos lenguas objetivos en un

mismo corpus, el entrenamiento se realiza como en un sistema clásico TAN. Por otra parte, la etapa de decodificación sigue los mismos lineamientos que la TAN.

3.2. Traducción automática multilingüe con compartición de parámetros controlados en NMT-Keras

Debido a que NMT-Keras está diseñado para construir modelos de traducción de una lengua a otra (Figura 1.2), fue necesario modificar varios aspectos del toolkit a nivel de decodificadores y entrenamiento para obtener una representación como la presentada en la Figura 2.3.

Como se exhibió en la Figura 3.1, NMT-Keras está basado en el modelo atencional de Bahdanau [9]. Por lo anterior, los esfuerzos de modificar el toolkit[1] fueron puestos desde la capa de salida del codificador, la cual fue conectada a dos decodificadores en paralelo, con el fin de separar los lenguajes en la etapa de decodificación. Lo anterior indica que se mantuvo el codificador original del toolkit, y por otra parte, se replicaron los tensores que construyen el decodificador. De esta forma queda un modelo constituido por dos submodelos:

- Submodelo 1: Codificador1 - Decodificador1
- Submodelo 2: Codificador1 - Decodificador2

Al adquirir esta arquitectura y a diferencia del entrenamiento con parámetros totalmente compartidos, los corpus no fueron concatenados, sino que, fueron tratados paralelamente. Además, como en este caso cada decodificador trabaja con un lenguaje en específico, el conjunto de datos de cada lengua objetivo, fue dirigido a cada decodificador en paralelo.

Por otra parte, NMT-Keras construye un modelo para la etapa de decodificación, el cual fue replicado para cada lengua. Dado lo anterior, es importante recordar en qué decodificador fue asignada cada lengua, debido a que en la etapa de traducción se debe saber en cual submodelo fue asignada la lengua a traducir.

En la Figura 3.2 se pueden observar 6 lineas salientes del modelo. Estas lineas son los tensores pertenecientes a 3 capas:

- Anotaciones (annotations layer)
- Memoria inicial (Initial memory layer)
- Estado inicial (Initial state layer)

[1] `https://github.com/cokecuevas/nmt-keras.git`.

En teoría, si existiesen N lenguas, se tendrían N decodificadores, por lo tanto, cada capa mencionada anteriormente presentarían N salidas. En este caso, como hay solo dos lenguas de destino, hay 2 salidas por cada capa las cuales conectan a cada decodificador como se ve en la Figura 3.2. Por otra parte, en la misma figura se puede distinguir una cuarta entrada a los decodificadores. Esta entrada es llamada "state_below", la cual representa al lenguaje objetivo. Es justamente esta capa la que permite distinguir entre ambos decodificadores al momento de dirigir el lenguaje objetivo al entrenar el modelo.

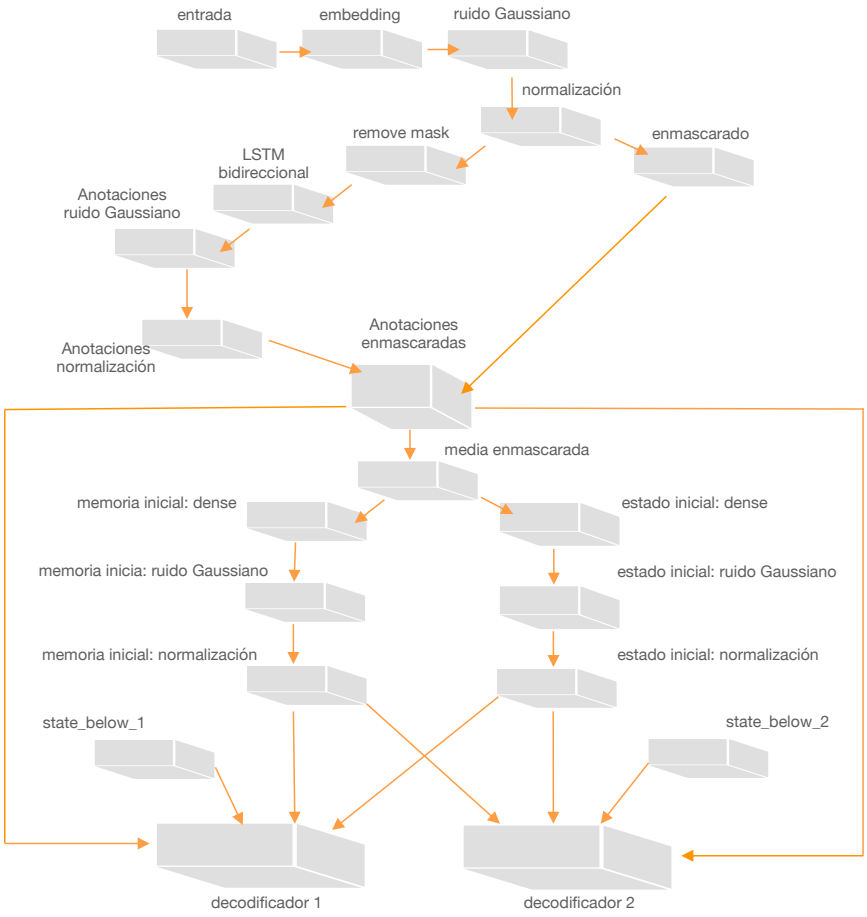

Figura 3.2: Configuración resultante en NMT-Keras para que permita la traducción de una lengua a dos lenguas.

Preproceso y Entrenamiento

El preprocesamiento es idéntico al aplicado al modelo con parámetros completamente compartidos. Sin embargo, el entrenamiento y traducción cambia de forma drástica.

Al poseer un modelo compuesto por submodelos, es necesario distribuir el entrenamiento en dos fases. La primera fase entrenará durante una época al submodelo 1 (inglés a lengua objetivo 1) y la segunda fase entrenará al submodelo 2 (inglés a lengua objetivo 2) durante una época.

La idea de distribuir el entrenamiento en dos fases es homogeneizar el modelo, dado que el codificador recibirá información de dos conjuntos de datos distintos. Por otra parte, este es el camino más sencillo para modificar al mínimo NMT-Keras. El proceso descrito se muestra en la siguiente imagen:

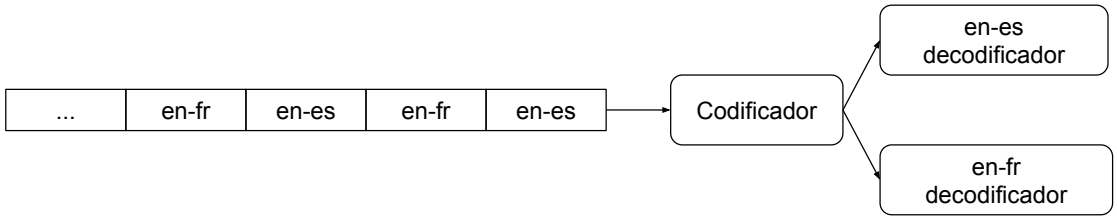

Figura 3.3: Entrenamiento del modelo propuesto en NMT-Keras para el caso de la traducción del inglés al español-francés.

Cada submodelo posee llamadas de vueltas (o "callbacks" en inglés) personalizadas, con el fin de controlar las métricas e hiperparámetros.

El tiempo de entrenamiento de este modelo se extiende bastante más que un clásico modelo de traducción automática, debido a que las etapas de entrenamiento no se ejecutan en paralelo. Aunque un entrenamiento utilizando multitareas [29] podría ser implementado para solucionar esta lentitud.

Capítulo 4

Marco experimental y Resultados

En las siguientes secciones se abordará el marco experimental, el cual será aplicado con los corpus inglés-español e inglés- francés, con la finalidad de encontrar la mejor configuración posible para la construcción de los modelos multilingües. Seguidamente, en la sección de resultados, se aplicarán estas configuraciones a todos los corpus, incluyendo el inglés-alemán, y sus combinaciones, tales como inglés-español:alemán e inglés-francés:alemán.

4.1. Marco experimental

En el presente documento se desarrollaron dos lineas de trabajo dentro de la traducción automática multilingüe: TAM con parámetros totalmente compartidos y TAM con parámetros controladamente compartidos. En ambas lineas se utilizaron los mismos corpus. Análogamente se desarrollaron experimentos en paralelo los cuales poseían ciertas características comunes. El objetivo del marco experimental es encontrar el mejor modelo dentro de distintas configuraciones e hiperparámetros utilizando un corpus reducido. El corpus utilizado para la selección de la mejor configuración fue el inglés como lenguajes fuente, y el francés y español como lenguaje destino. Seguidamente estos modelos se entrenaron con la totalidad de los datos disponibles para todas las lenguas de destino (español, francés y alemán). Seguidamente se presenta el comportamiento de cada modelo entrenado en el apartado de resultados.

Ambas partes de este capítulo fueron desarrolladas con el toolkit NMT-Keras y las modificaciones que se realizaron sobre este para lograr el entrenamiento de múltiples lenguas en un único modelo. Estas modificaciones fueron introducidas en la Sección 3.2.

Corpus

Los corpus utilizados en este trabajo son los pertenecientes al Europarl, el cual posee 21 lenguas europeas, de las cuales se utilizaron 3. Por otra parte, estos corpus fueron utilizados en la conferencia "Machine Translation"(WMT).

En este trabajo se realizaron traductores automáticos multilingües en NMT-Keras para una lengua de origen y dos lenguas de destino. La lengua de origen es el inglés (para todas las lenguas de destino), mientras que las lenguas de destino son español, francés y alemán. Todos los corpus poseen sus frases pareadas: inglés-español, inglés-francés e inglés-alemán. En la Tabla 4.1 se muestra el tamaño de los corpus.

Lenguaje	En-Es	En-Fr	En-Al
# Sentencias	1.965.734	2.007.723	1.813.092
# Palabras fuente	56.801.281	58.073.788	53.073.122
# Palabras destino	61.907.970	66.277.191	55.760.756

Tabla 4.1: Tamaño del corpus de entrenamiento para todas las lenguas de destino.

Por otra parte el corpus para desarrollo pertenece a la misma conferencia.

Lenguaje	En-Es	En-Fr	En-Al
# Sentencias	3.003	3.003	2998
# Palabras fuente	63.779	62.338	67.453
# Palabras destino	69.471	69.631	64.031

Tabla 4.2: Tamaño del corpus de validación para todas las lenguas de destino.

Finalmente se tiene el corpus de prueba, el cual pertenece a la conferencia del año 2014 para el español, 2015 para el francés y 2020 para el alemán.

Lenguaje	En-Es	En-Fr	En-Al
# Sentencias	3.000	1.500	2.000
# Palabras fuente	56.089	23.668	39.368
# Palabras destino	62.045	25.147	35.983

Tabla 4.3: Tamaño del corpus de prueba para todas las lenguas de destino.

Entrenamiento

Las métricas de entrenamiento se ajustaron basadas en el largo periodo de entrenamiento que requiere un traductor automático de este tamaño, además de los recursos computacionales que se poseían para realizar este trabajo.

Dado lo anterior, se configuró un entrenamiento de 100 épocas por cada experimento, con un paciencia de 10. El tamaño de las sentencias de entrada y salida para todos los experimentos fue configurada en 70. Al tamaño de la búsqueda se le asignó un valor de 6.

Cabe destacar que todos los experimentos fueron realizados sobre un codificador constituido por capas de LSTM y el decodificador compuesto por capas del tipo Conditional LSTM.

Linea base

Para realizar los experimentos se utilizó el 30 % de los datos de entrenamiento disponibles del corpus inglés-español e inglés-francés, con el fin de acelerar la elección de la mejor configuración. En la Tabla 4.4 se puede observar la reducción del corpus. Con los datos disponibles en esta etapa, se realizó la búsqueda de una configuración

Lenguaje	En-Es	En-Fr
# Sentencias	589.720	602.316
# Palabras fuente	17.238.644	17.672.387
# Palabras destino	18.606.704	20.108.162
Tam. Vocabulario fuente	17.766	17.808
Tam. Vocabulario destino	18.542	18.357

Tabla 4.4: Tamaño del corpus reducido para la linea base.

para encontrar la mejor configuración que logre representar la linea base para todas las lenguas de este estudio.

Dentro de una serie de diversas configuraciones de las cuales se variaron algoritmos de entrenamiento, tamaño del lote de entrenamiento, tamaños de los embeddings, entre otros. Se buscó aquella configuración que presentara el más alto índice BLEU.

La tasa de aprendizaje se fijó en 0.0002 para el algoritmo de Adam y 1.0 para Adadelta. El tamaño del lote o en inglés "batch size" fue configurado en 26 para las configuraciones con tamaño de 1000 en el codificador y decodificador, y para las redes de 512 y 600 el tamaño del lote asignado fue de 60. No se realizaron experimentos con tamaños de lote mayores a 60, puesto que, se le dio prioridad al tamaño del vocabulario al momento de realizar el entrenamiento, por lo tanto, en este trabajo se prefirió disminuir el tamaño del lote y mantener un vocabulario amplio.

En la Tabla 4.5 se pueden ver algunas de las configuraciones probadas.

Exp.	Algoritmo	Emb. fuente	Emb. objetivo	#Capas codificador	#Capas decodificador	Tamaño codificador	Tamaño decodificador
Español							
1	Adam	600	600	2	2	1000	1000
2	Adam	512	512	1	1	512	512
3	Adadelta	512	512	1	2	512	512
Francés							
1	Adam	600	600	2	2	1000	1000
2	Adam	512	512	1	1	512	512
3	Adadelta	512	512	1	2	512	512

Tabla 4.5: Mejores configuraciones encontradas.

Dentro de las configuraciones presentadas en la Tabla 4.5, se seleccionó aquella que mayor puntaje BLEU presentara por cada lengua. Por lo que el BLEU máximo alcanzado para el español fue de 22,25 utilizando la configuración del experimento 2, mientras que para el francés fue de 21,61 utilizando la configuración del experimento 2. Por lo que la linea base para la etapa experimental del trabajo se presenta en la Tabla 4.6.

Lenguaje fuente	Lenguaje destino	BLEU	TER(%)
Inglés	Español	22,2	59,5
	Francés	21,6	64,0

Tabla 4.6: Linea base obtenida sobre el conjunto de prueba, para el español y francés utilizando un corpus reducido.

En la Figura 4.1 se puede observar el comportamiento del entrenamiento de cada lengua para la linea base.

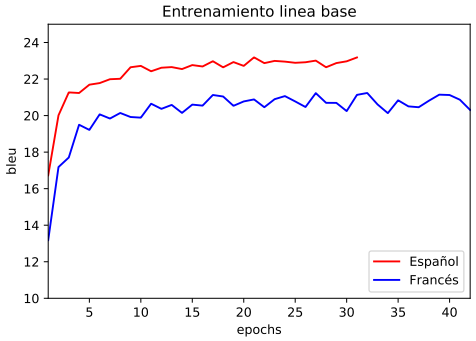

Figura 4.1: BLEU por epoch resultante para la linea base. Una epoch equivale a la presentación completa de las muestras de entrenamiento por cada lengua.

Basado en los resultados anteriores, se presentará en la sección de resultados el comportamiento de esta configuración con la totalidad de los datos y para todas las lenguas abarcadas en este estudio. Por lo que obtendrán nuevas lineas bases por pares de lengua.

Experimentos TAM con parámetros totalmente compartidos

Para realizar los experimentos para la TAM con parámetros totalmente compartidos, se siguieron los lineamientos expuestos en la linea base. Se realizaron los mismos experimentos con las mismas configuraciones. Sin embargo, existe una variación en este experimento a nivel de corpus, debido a que los corpus fueron concatenados como se expuso en la Sección 3.1. Por otra parte, este tipo de aproximación no requiere modificación a nivel de código, debido a que la detección del lenguaje de destino se realiza por medio del etiquetado del corpus del lenguaje fuente, tal como se exhibió en la Figura 2.2.

Como este tipo de aproximación realiza la concatenación del corpus fuente y destino, el tamaño del corpus aumenta drásticamente, como también los vocabularios. En la Tabla 4.7 se muestra el tamaño del corpus para el conjunto de datos utilizado en esta etapa del trabajo.

Lenguaje	En-Es + En-Fr
# Sentencias	1.192.036
# Palabras fuente	36.102.005
# Palabras destino	38.714.866
Tam. Vocabulario fuente	17.871
Tam. Vocabulario destino	26.979

Tabla 4.7: Tamaño del corpus de entrenamiento para TAM con parámetros totalmente compartidos.

Dentro de las mejores configuraciones exhibidas en la Tabla 4.5, la que mejor BLEU presentó para este tipo de tarea, fue la configuración número 2 la cual entregó un BLEU de 21.2 para el español y 21.0 para el francés, tal como se muestra en la siguiente tabla:

Lenguaje fuente	Lenguaje destino	BLEU	TER(%)
Inglés	Español	21,2	62,8
	Francés	21,0	66,7
	Español:Francés	20,9	64,0

Tabla 4.8: BLEU y TER obtenido sobre el conjunto de prueba, con el corpus reducido para el español, francés y ambos corpus concatenados.

En la Figura 4.2 se observa el comportamiento del modelo con parámetros totalmente compartidos durante el entrenamiento.

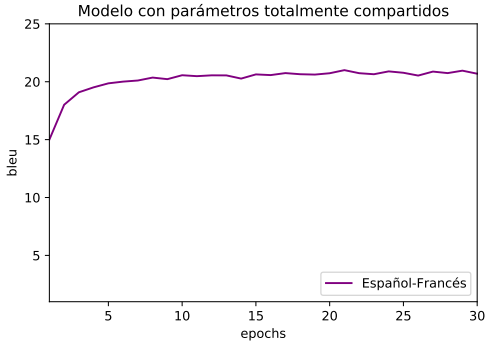

Figura 4.2: BLEU por epoch resultante para el modelo con parámetros totalmente compartidos. Una epoch equivale a la presentación completa de las muestras de entrenamiento para ambas lenguas unidas.

Experimentos TAM con parámetros controladamente compartidos

Como se mencionó en la Sección 3.2, fue necesario modificar varios aspectos de NMT-Keras para que sea posible la ejecución de este experimento. Sin embargo, aunque los corpus y los parámetros de entrenamiento sean comunes, en este caso el entrenamiento tuvo variaciones a nivel de ejecución tal como se menciona en la Sección 3.2.

Lo anterior no implica que se deban realizar pruebas distintas a las realizadas en la linea base y parámetros completamente compartidos, por lo que se realizaron exactamente las mismas pruebas y configuraciones con tal de encontrar aquel modelo que mejor calidad de traducción entregue. Por lo tanto, siguiendo los mismos lineamientos que los modelos anteriores se probaron diversas configuraciones, siendo la mejor la que posee el algoritmo de Adam y los embeddings de tamaño 512.

Como en esta aproximación los decodificadores están separados y el entrenamiento se realizó con submodelos, tal como se muestra en la Figura 3.3, se utilizaron ambos conjuntos de datos paralelizados, los cuales se describieron en la Tabla 4.4.

El comportamiento del modelo seleccionado se muestra en la Figura 4.3

El BLEU alcanzado en esta etapa para esta aproximación fue de 23,7 para el español y 23,6 para el francés. En la siguiente tabla se resumen los resultados obtenidos:

Lenguaje fuente	Lenguaje destino	BLEU	TER(%)
Inglés	Español	23,7	57,0
	Francés	23,6	60,0

Tabla 4.9: BLEU y TER obtenido sobre el conjunto de prueba, con el corpus reducido para el español y el francés con un decodificador para cada lengua.

Cabe destacar que en esta aproximación no existe un BLEU estimado para el español-francés en conjunto, puesto que en esta parte de la experimentación se cons-

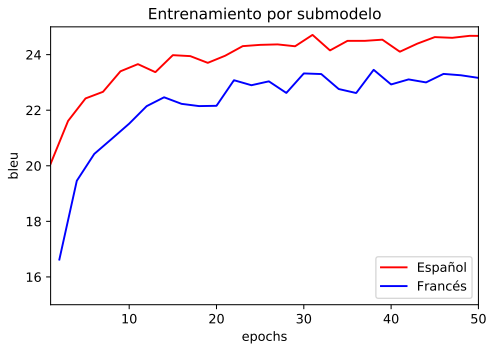

Figura 4.3: BLEU por epoch resultante para el modelo con parámetros controladamente compartidos. Una epoch equivale a la presentación completa de las muestras de entrenamiento por cada submodelo.

truyó un modelo con un decodificador para cada lengua.

4.2. Resultados

En la sección anterior se expusieron los mejores modelos encontrados con los corpus reducidos. En esta parte del estudio se presentarán los resultados obtenidos con los modelos seleccionados anteriormente para cada aproximación y cada lengua de destino, incluyendo la linea base y utilizando la totalidad de los datos, los cuales se describen en la Tabla 4.1.

Linea base

Para poder comprobar los beneficios de la traducción automática multilingüe en NMT-Keras, fue necesario establecer una linea base con los métodos tradicionales de TAN utilizando RNN. Para esto, al igual que en la sección anterior, se entrenó un modelo en NMT-Keras, para el español, un modelo para el francés y un modelo para el alemán. Estos modelos fueron entrenados con la totalidad de los datos disponibles, los cuales se describieron en la Tabla 4.1. La linea base resultante se presenta en la Tabla 4.10.

Lenguaje fuente	Lenguaje destino	BLEU	TER(%)
Inglés	Español	23,6	58,0
	Francés	24,7	60,0
	Alemán	15,0	68,3

Tabla 4.10: BLEU y TER obtenido sobre el conjunto de prueba, con el corpus completo para el español, francés y alemán.

El tiempo total de entrenamiento para el español fue de 9,3 días, mientras que para el francés 12,7 días y para el alemán 9,5 días.

La cantidad de parámetros entrenables para el español fue de una cantidad de 43.861.345, para el francés 43.787.009 y para el alemán 47.836.961.

Experimentos TAM con parámetros totalmente compartidos

Los resultados obtenidos en el entrenamiento de TAM con parámetros totalmente compartidos con la totalidad de los datos en NMT-Keras, se muestran en la siguiente tabla:

Lenguaje fuente	Lenguaje destino	BLEU	TER(%)
Inglés	Español	21,3	61,0
	Francés	22,4	63,0
	Español:Francés	23,2	60,0
Inglés	Español	22,5	60,0
	Alemán	14,4	73,6
	Español:Alemán	19,0	64,8
Inglés	Francés	22,4	63,8
	Alemán	13,5	74,2
	Francés:Alemán	17,3	68,9

Tabla 4.11: BLEU y TER obtenido sobre el conjunto de prueba, con la totalidad del corpus para el español francés y alemán. El BLEU y TER de los lenguajes de destino Español:Francés, Español:Alemán y Francés:Alemán, son calculados sobre el corpus concatenado de ambas lenguas.

Se puede observar de la Tabla 4.11, que esta aproximación no mejora la calidad de la traducción. Aunque ambos modelos utilizan la misma arquitectura, según estos resultados la TAN multilingüe con parámetros totalmente compartidos no mejora la traducción, aunque logra diferenciar ambos lenguajes gracias al etiquetado del corpus fuente.

La cantidad de parámetros entrenables para este modelo fueron de un total de 51.840.424, mientras que la duración del entrenamiento fue durante alrededor de 17 días.

Experimentos TAM con parámetros controladamente compartidos

Los resultados obtenidos con la totalidad del corpus para esta aproximación se presentan en la Tabla 4.12.

De la tabla, se infiere que este tipo de aproximación logra obtener un BLEU superior de casi 2 puntos de BLEU en la traducción del inglés al español y del inglés al francés con respecto a la linea base (ver Tabla 4.10). Lo anterior, independientemente de la combinación de lenguas que se haya utilizado en los submodelos. En la misma

Lenguaje fuente	Lenguaje destino	BLEU	TER(%)
Inglés	Español	25,4	57,0
	Francés	26,5	59,0
Inglés	Español	25,4	56,9
	Alemán	17,0	68,0
Inglés	Francés	26,6	59,0
	Alemán	16,8	68,6

Tabla 4.12: BLEU y TER obtenido sobre el conjunto de prueba, con la totalidad del corpus para el español, francés y alemán

tabla, se observa que en la traducción del inglés a alemán se supera la linea base en casi 2 puntos de BLEU en ambas combinaciones de submodelos. Por otra parte, este modelo supera por alrededor de 4 puntos en la misma tarea a la TAM con parámetros completamente compartidos.

El tiempo de entrenamiento de este tipo de modelo se calcula con el sumatorio del tiempo de cada submodelo. Por ejemplo: el submodelo el cual controla el español tardó 5,37 horas por cada época ejecutada, mientras que el submodelo perteneciente al francés tardó 5,46 horas. De lo anterior se puede deducir que el tiempo de entrenamiento de una época del modelo completo es de 11,38 horas. Este modelo fue entrenado durante 100 épocas (50 para el español y 50 para el francés), lo que da un total de 24,7 días de entrenamiento. Paralelamente, los modelos entrenados con los pares de lenguas español-alemán y francés-alemán, tuvieron un tiempo de entrenamiento similar al tiempo del español-francés.

Por otra parte, la cantidad de parámetros entrenables para el modelo de traducción de inglés a español-francés se calculó basado en la cantidad de parámetros entrenables de cada submodelo. Por lo tanto, se tiene que el submodelo del español tuvo un total de 43.885.409 parámetros entrenables, mientras que para el francés 43.787.009. Cabe destacar que ambos submodelos compartían el mismo codificador, por lo que la totalidad de parámetros del modelo completo es calculado como el sumatorio de los parámetros presentados por el codificador más los parámetros presentados por cada decodificador por separado. En este escenario se tiene que el codificador presentó una totalidad de 25.841.152 parámetros. Por lo tanto, la totalidad de parámetros presentados por esta aproximación fue de un total de 61.831.266.

En cuanto al modelo de traducción de inglés a español-alemán, este presenta una totalidad de 63.835.029 parámetros entrenables.

Finalmente, el modelo de traducción de inglés a francés-alemán, presentó una totalidad de 65.933.602 parámetros entrenables.

Discusión de resultados

De los resultados expuestos anteriormente se puede observar que la TAM con parámetros totalmente compartidos presentó un BLEU de 21,32 para el español y

22,43 para el francés en el modelo que traduce el inglés al español y francés. Este resultado se acerca a la linea base la cual obtuvo 23,65 en el español y 24,78 en el francés. Sin embargo, es evidente que este tipo de aproximación no logra superar a la linea base en este experimento. Sumado a lo anterior, los resultados expuestos por los modelos de traducción de inglés a español-alemán y francés-alemán, tampoco logran mejorar la linea base de cada una de sus lenguas, aunque, en el caso del alemán se acerca bastante.

Por otra parte, en los experimentos realizados con la TAM con parámetros controladamente compartidos, el modelo que traduce del inglés al español-francés, presentó un BLEU de 25,48 en español, superando en casi 2 puntos de BLEU a la linea base en este experimento. Este modelo se acerca bastante a resultados presentados en la literatura. En Dong et at. [29], se presenta un modelo el cual utiliza 4 lenguajes objetivos (español, francés, portugués y holandés). Este modelo presenta un BLEU de 25,31 en la traducción del inglés al español para el mismo conjunto de pruebas, obteniendo 0,17 puntos de BLEU menos que el modelo con parámetros controladamente compartidos presentado en este trabajo. Al mismo tiempo, en el documento se expone un BLEU de 23,58 obtenido en MOSES para este conjunto de pruebas, el cual es superado por el modelo propuesto. En cuanto al francés, se obtuvo un BLEU de 26,58, el cual es un puntaje aceptable para la reducida configuración utilizada, sin embargo, comparando esta aproximación con las presentadas en la conferencia de la WMT15 [41], el modelo propuesto para el francés, no logra posicionarse entre los mejores, los cuales obtuvieron entre 30 y 34 puntos de BLEU.

Observando los resultados que se obtuvieron con los modelos de traducción de inglés a español-alemán, se estima que la combinación entre estas dos lenguas en la aproximación presentada, logra mejorar la calidad de la traducción. Sumado a lo anterior, el modelo de traducción de inglés a francés-alemán obtuvo una mejora significativa al igual que los modelos presentados en este tipo de modelo.

En todos los casos el alemán de por si no mejora. Si a este se le añade una lengua como el francés o español empeora con respecto a su linea base. Sin embargo, el francés y español combinado con el alemán mejoran al igual que si combinamos francés y español, incluso obteniendo valores de BLEU y TER casi idénticos. La diferencia radica en que el francés y español mejoran en conjunto si se posicionan en un mismo modelo. En su caso caso contrario, si se combina una de estas lenguas con el alemán solo se beneficia una lengua y no ambas.

Se estima que lo anterior sucede porque al compartir el mismo codificador y la misma lengua fuente, se enriquece la generalización y el vocabulario en el codificador. Pero en la etapa de decodificación el alemán con el francés no generan una compartición de conocimiento, como tampoco, el alemán con el español. Esto es debido a que son lenguas con bases lingüisticas distantes. Esto se acentúa más cuando separamos los decodificadores, ya que cada decodificador se preocupa de cada lengua, además, ambas lenguas reciben una representación del codificador que fue entrenado con la misma lengua con casi el doble de la cantidad de ejemplos que se entrenó en la linea base. Por lo anterior, se estima, que al separar los decodificadores, el francés y el español se ven beneficiados por el rico entrenamiento que

recibió el codificador, incluso el alemán en este punto aumenta 3-5 puntos más que los parámetros totalmente compartidos.

Cabe mencionar que en este estudio se aplicó la prueba de "approximate randomization tests" (Riezler and Maxwell, 2005 [42]) a los modelos que comparten lenguas de destino. La prueba fue configurada con 10.000 repeticiones y un p=0.05, con el fin de determinar que los modelos presentan diferencias estadísticamente significativas.

Análogamente, de los resultados se observa que la cantidad de parámetros aumenta en un 16 % en el modelo con parámetros totalmente compartidos y en un 30 % en el modelo con parámetros controladamente compartidos. Seguidamente, los datos indican que el tiempo de entrenamiento de un modelo multilingüe es de hasta un 50 % más que un modelo de TA tradicional.

Finalmente, comparando ambas aproximaciones de TAM presentadas en este trabajo, se observó que para esta tarea la TAM con parámetros controladamente compartidos presenta una mejor calidad de traducción, frente a la TAM con parámetros completamente compartidos, superando a esta aproximación en 4 puntos de BLEU. Lo anterior demuestra que al poseer un decodificador para cada lengua aumenta la calidad de la traducción, puesto que el decodificador se especializa en la traducción de un solo lenguaje y no varios.

Capítulo 5

Trabajos Futuros y Conclusiones

5.1. Trabajos Futuros

Incorporar nuevas funcionalidades multilingües a NMT-Keras, como el entrenamiento de un lenguaje a muchos, muchos lenguajes a uno y muchos lenguajes a muchos lenguajes. De esta forma, sería posible la parametrización de los lenguajes a utilizar en los modelos de traducción automática. Además de esto, añadir la posibilidad de decidir qué parámetros dentro de los componentes del modelo pueden ser compartidos y cuales no. Lo anterior, no aplicado tan solo a la configuración utilizada en este trabajo, sino que también aplicada al transformer. De esta forma se podría estudiar la compartición de distintos parámetros dentro de los decodificadores con una simple configuración como lo hace NMT-Keras para la traducción automática clásica.

Buscar métodos más eficientes para el entrenamiento de modelos con parámetros parcialmente compartidos, puesto que, el entrenamiento de estos modelos tarda bastante más que un entrenamiento de un traductor automático no multilingüe, además, es sabido que entrenar este tipo de modelos no es un problema trivial, es por esto que es necesario optimizar el entrenamiento.

Finalmente, estudiar qué lenguas son las mejores para mejorar la calidad de la traducción de otras lenguas.

5.2. Conclusiones

Los resultados muestran que la traducción automática multilingüe mejora la calidad de la traducción si se utiliza un segundo lenguaje con algún grado de similitud lingüístico. Sin embargo, esta afirmación depende del diseño del modelo multilingüe, puesto que en este trabajo se demostró que la TAM con parámetros totalmente compartidos no supera la linea base. Aunque esto podría ser justificado por no utilizar una configuración orientada al decodificador. Lo anterior quiere decir, que si se comparan las 3 aproximaciones expuestas en este trabajo, se puede observar que cada una de ellas utilizan la misma configuración, sin embargo, el modelo con todos los parámetros compartidos tiene el trabajo extra de procesar múltiples lenguajes en un mismo decodificador con un tamaño definido. Esto quiere decir, que si se hubiese puesto más atención a la configuración del decodificador (aumentar/disminuir tamaños de configuraciones) se podría haber obtenido mejores resultados.

La separación de los decodificadores por lengua aumentó drásticamente la calidad de la traducción, esto indica que al poner mayor atención a los decodificadores paralelos se logran mejores traducciones. Sin embargo, en este trabajo se probaron diversas configuraciones dentro del límite de recursos computacionales que se poseían. El aumentar el tamaño de las capas de la redes neuronales y los tamaños de los embeddings podría haber mejorado la calidad de la traducción, pero a su vez, afecta directamente a los recursos disponibles para entrenar los modelos, es por lo anterior que no se probaron configuraciones más grandes a nivel de codificadores y decodificadores, además, el tiempo de entrenamiento que implica esta variación. Como se demostró en este trabajo, el entrenamiento de este tipo de aproximación mejora la calidad de la traducción pero a un costo de tiempo bastante mayor a un entrenamiento común de un traductor automático bilingüe.

La complejidad de cada aproximación en la TAM presentada en este trabajo fue puesta en evidencia sobre el toolkit NMT-Keras. Se demostró que el entrenamiento de un TAM con parámetros totalmente compartidos es sencillo, debido a la adición de la etiqueta del lenguaje destino en las sentencias de entrada, lo que permitió entrenar un modelo multilingüe de forma rápida sobre el toolkit. En su contraparte, en la TAM con parámetros controladamente compartidos fue necesario modificar una serie de componentes en NMT-Keras, los cuales permitieron construir la configuración del modelo (ver Figura 3.2). Esta modificación no solo se traduce en una variación a nivel de la configuración del modelo, sino que también, la creación de un método de entrenamiento alternado (ver Figura 3.3) que el toolkit no poseía, además de la detección del lenguaje a traducir en la etapa de muestreo, como también, en la etapa de traducción del conjunto de pruebas. Lo anterior demuestra la alta complejidad que está presentado la TAM al compartir los parámetros de los distintos componentes de los modelos.

Bibliografía

[1] D. Ortiz-Martínez. Advances in Fully-Automatic and Interactive Phrase- Based Statistical Machine Translation. PhD thesis, Universidad Politécnica de Valencia. 2011. Advisors: Ismael García Varea and Francisco Casacuberta.

[2] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra,and R. L. Mercer. 1993. The mathematics of machine translation: Parameter estimation.Computational Linguistics, 19(2):263–311.

[3] F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine trans-lation. 2002. In Proceedings of the 40th Annual Meeting on Association for ComputationalLinguistics, páginas 295–302.

[4] Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. arXiv:1609.08144

[5] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), páginas 1-61.

[6] Cho, K., van Merrienboer, B., Gulçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. 2014. Learning phrase representations using RNN encoderdecoder for statistical machine translation. CoRR, abs/1406.1078.

[7] Sutskever, I., Vinyals, O., and Le, Q. 2014. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems. arXiv:1409.3215.

[8] Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. 2003. "A neural probabilistic language model". Journal of Machine Learning Research, v. 3, páginas 1137–1155.

[9] Bahdanau, D., Cho, K., and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. Technical report, arXiv:1409.0473.

[10] Cho, K., van Merrienboer, B., Gulçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. 2014. Learning phrase representations using RNN encoderdecoder for statistical machine translation. CoRR, abs/1406.1078.

[11] Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. 2017. A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), páginas 1925–1935.

[12] Yun Chen, Yang Liu, and Victor O. K. Li. 2018. Zero-Resource Neural Machine Translation with Multi-Agent Communication Game. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. AAAI Press, páginas 5086–5093.

[13] Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint Training for Pivot-based Neural Machine Translation. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. Melbourne, páginas 3974–3980.

[14] Raj Dabre, Chenhui Chua, Anoop Kunchukuttan. 2020. Comprehensive Survey of Multilingual Neural Machine Translation. arXiv:2001.01115v2

[15] Barret Zoph and Kevin Knight. 2016. Multi-Source Neural Translation. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, páginas 30–34.

[16] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering 22, páginas 1345–1359.

[17] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. Transactions of the Association for Computational Linguistics 5, páginas 339–351.

[18] Graham Neubig and Junjie Hu. 2018. Rapid Adaptation of Neural Machine Translation to New Languages. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, páginas 875–880.

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of RareWords with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, 1715–1725. http://www.aclweb.org/anthology/P16-1162

[20] Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. International Conference on Acoustics, Speech, and Signal Processing, IEEE, páginas 5149–5152.

[21] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, páginas 66–71.

[22] Devendra Sachan and Graham Neubig. 2018. Parameter Sharing Methods for Multilingual Self-Attentional Translation Models. In Proceedings of the Third Conference on Machine Translation: Research Papers. Association for Computational Linguistics, páginas 261–271.

[23] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), páginas 1723–1732.

[24] Castaño, M. A., Casacuberta, F., and Vidal, E. 1997. Machine translation using neural networks and finite-state models. Theoretical and Methodological Issues in Machine Translation (TMI), páginas 160–167.

[25] Álvaro Peris, Francisco Casacuberta. 2018. NMT-Keras: a Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning. The Prague Bulletin of Mathematical Linguistics. arXiv:1807.03096

[26] Chollet, François et al. Keras. https://github.com/keras-team/keras. 2015. GitHub repository. 121 PBML 111 OCTOBER 2018 Duchi, John, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12:2121–2159, 2011.

[27] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017.Attention Is All You Need. In Proceedings of NIPS, volume 30, páginas 5998–6008.

[28] Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting Multilingualism through Multistage Fine-Tuning for Low-Resource Neural Machine Translation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), páginas 1410–1416.

[29] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu and Haifeng Wang. 2015. Multi-Task Learning for Multiple Language Translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), páginas 1723-1732.

[30] Jordan, M. I. Artificial neural networks. 1990. chapter Attractor Dynamics and Parallelism in a Connectionist Sequential Machine, páginas 112–127.

[31] Elman, J. L. Finding structure in time. Cognitive science, 14(2):179–211.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. CoRR, abs/1706.03762.

[33] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. 2015. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, páginas 1412–1421.

[34] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

[35] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), páginas 1724–1734.

[36] Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, Lei Li. 2020. Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information. arXiv:2010.03142v1.

[37] Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. CoRR, abs/1606.08415.

[38] Markus Freitag and Orhan Firat. 2020. Complete Multilingual Neural Machine Translation. arXiv:2010.10239v1.

[39] Luyu Gao, Xinyi Wang, and Graham Neubig. 2020. Improving Target-side Lexical Transfer in Multilingual Neural Machine Translation. arXiv:2010.01667v1

[40] Xinyi Wang and Graham Neubig. 2019. Target conditioned sampling: Optimizing data selection for multilingual neural machine translation.

[41] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia and Marco Turchi.

2015. Findings of the 2015 Workshop on Statistical Machine Translation. Proceedings of the Tenth Workshop on Statistical Machine Translation, páginas 1-46.

[42] Riezler, S. and Maxwell, J. T. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, páginas 57–64.