uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería en Tecnologías de la Telecomunicación

2018-2019

Trabajo Fin de Grado

"Modelado de lenguaje natural con aprendizaje profundo"

Carlos Herrera Díaz

Tutor

Pablo Martínez Olmos

Leganés 24/9/2018

“LA CIENCIA ES UN MITO, SÓLO QUE ES EL MITO MÁS HERMOSO, EL ÚNICO GENERALIZABLE A TODA LA ESPECIE Y QUIZÁS EL MÁS DIGNO DE RESPETARSE.”

ANTONIO ESCOHOTADO ESPINOSA

AGRADECIMIENTOS

Esta etapa de mi corta vida ha sido la más enriquecedora y llena de experiencias hasta la fecha. Con este trabajo, en cierta medida, pongo un simbólico broche final a ella y no quiero olvidarme de las personas que me han acompañado e impulsado en ella y siempre. Gracias en especial a mi familia por apoyarme, soportarme y concederme las oportunidades de las que gozo para desarrollarme como persona. Tampoco quiero olvidarme de la familia por elección, grandes amigos de los que me han dejado apoyarme para constituirme en cierta medida en lo que soy hoy. Mención especial para Fernando, por estar siempre ahí que es más de lo que se puede pedir. Ojalá que en las etapas por venir las personas que me han acompañado hasta ahora continúen conmigo.

ABSTRACT

From some years now machine learning, big data, data analysis… have been recurrent and trendy words in the IT world and general public. The computation power increase has given the chance to widespread the use of machine learning algorithms to almost every aspect in our daily basis. This kind of algorithms are present from the recommendation system of your favourite social media app, to the new cars with autopilot that in future will change our concept of transport.

This project is focused in the problematics of natural language translation. Natural language translation is nowadays a well known tool available for everybody in the developed world. It is common for people to use this kind of tools like Google Translate, DeepL Translator… for multiple tasks at work, while studying, or just when having doubts about any kind of translation. The development of this technology has improved the communication between people making it easier, faster and cheaper even if the user does not have any knowledge of the language he is translating to.

The main aim of this project is to dissect and understand the broadly used architecture known as sequence to sequence, a coder-decoder structure. Making a good translator is not an easy task, computational power is the main obstacle to face in our case, "We found that the large model configuration typically trains in 2-3 days on 8 GPUs using distributed training in Tensorflow."[1] obviously I did not have the computing resources as the developers referred so creating a good translator is not an achievable goal for me. However, it is possible to make a translator not that good but which outputs interesting enough data in order to study its behaviour.

Problem analysis

Translating is a complex solving problem task as grammatical structures, pronouns, absence of a literal translation, vocabulary... differ from one language to another. This complexity means that this kind of problematic can not be tabulated or simplified to be a word by word translation or a searching algorithm.

One of the problems found is that we do not know beforehand the length of the output sentence, the length can vary from less, equal or more amount of words. Added to this, words can appear in different position or order making the translation a mixture of these words. To make a robust system the architecture must know de context of the sentence in order to give structure and sense. As an example to understand it: "Last night I ate an entire pizza" "Hier soir j'ai mangé une pizza entière".

[1] Denny Britz, Anna Goldie, Minh-Thang Luong, Quoc Le, "Tutorial: Neural Machine Translation"

We can identify that words do not have a literal translation “last night” “hier soir”(yesterday night). Just after that there is a grammatical structure not used in both languages, only present in french “j’ai”, which means a word contraction for *je ai* called liaison. Another complexity added is that french distinguishes between male *le* and female *la* in objects, on the other hand english is neutral using *the.* The analysis of this simple sentence is not over, we can see a clear word order change, “entire pizza” “pizza entière”.

Difficulties do not end in the logic of the problem, the amount of words of each language which our architecture will recognise must be limited, the more words known the more dimensions in computing are added. The length of the sentences also adds computational cost, so it is a matter also to take care of.

Sequence to sequence architecture

The sequence to sequence architecture can be characterized by two main parts, the coder and decoder. The aim of this architecture is to model de conditioned probability $p(y|x)$ of the translation of an input $x1, \ldots, xn$ sentence to a target sentence $y1, \ldots, yn$.

The input sequence is input in a recursive way into the coder, once computed the entry, the resultant information saved in the last cell state of the coder is used as input for the decoder. The decoder will compute this input and make recursive outputs until the system converges.

Structurally the architecture is made of two recursive neural networks which consist of basic long-short term memory cells concatenated one to another. Two independent embedding matrixes, one matrix is trained for one language and placed in the coder section, and the other matrix is trained for the other language and placed in the decoder section.

A projection layer is placed at the output of the decoder.

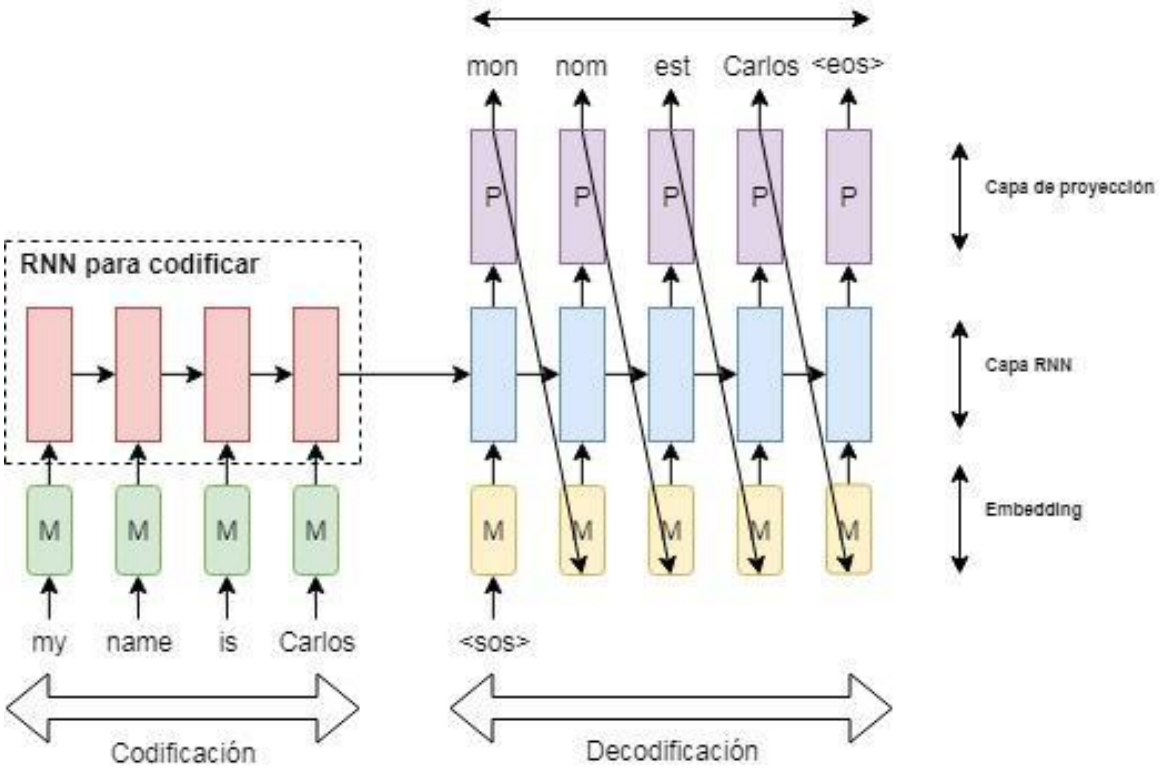

Fig 1

Different techniques are added to the coder-decoder system, embedding and a projection layer are fundamental in the logic of the problem (explained in 4.2 Embedding).

What the architecture is doing is a probability distribution of the vocabulary, it is the probability given the input sentence and the outputs achieved until that moment. This probability is represented as a vector which defines the vocabulary and the probability that that word is the next one. Choosing the correct word is a complex task and is analyzed in this project with two algorithms, greedy and beam search. Greedy selects the most probable output word while beam search expands it to a number of most probable words and chooses the best one in the end.

Problem solving

The data used in this problem requires to be a sufficient amount of sentences and to be a good translation between two languages. The data set used is a compound of orators interventions at the europarlament[2]. It is a complex text, with multiple references to names, places… and with lack of colloquial language. The similarity between languages

[2] "European Parliament Proceedings Parallel Corpus 1996-2011"

is an important aspect to take in account thus this issue has been studied and explained in 5.1 Pre procesado del texto.

In order to work with this kind of data several processes had to be made. This processes have as goal to normalize the text and make both languages have a similar approach to their abecedary.

After making this normalization, the data has been dissected to for example choose the words that defines the vocabulary known and unknown, the method used has been counting the appearance of every word throughout the whole text. After having the information of the number of appearances, the most repeated ones have been the chosen to represent the vocabulary known, the unknown ones have been represented and replaced in the original text as a well known token.

Due to the computational limitations, the validation process had as aim to give the system the enough flexibility to be adjusted to the amount of data used. In order to have a good translator it is necessary to use a large amount of different sentences, it has not been possible to do that so the goal has been to make a system that outputs interesting data that offers the possibility of studying it.

Computing the score is a complex issue, the way to do this in the project has been simple, comparing word by word between the output sentence and the target sentence. After comparing calculate a percentage of the words that represent the exact translation.

This way of scoring is in many cases too simple as there is not always an only way of translation, there are many different ways to understand and interpret sentences. Words can appear in different positions, different pronouns… and still make a good translation.

Knowing that the score given is going to be very pessimistic the way to read it is as a lower benchmark, and understand that the real score is significantly better.

The final score is about ¼ of the sentence correctly translated, which as said before, being a pessimistic view and knowing the computational limitations is a promising result.

To understand the problematics of scoring here it is an example:

Input	thank you very much commissioner
Output	merci beaucoup monsieur le commissaire
Target	merci infiniment madame la commissaire
Score	0.4

Analyzing the previous example we can clearly see the problematics at computing the score and how pessimistic it is. The output and the target sentence differ in very little amount of words, the difference between words are just that they both are synonyms

or the gender of the words are changed. English does not give information about the gender referred in the sentence and the translation has to asum it from scratch.

In terms of translation it can be said that the output is a good result and the system has behaved well in this case. A human computation of the score may be close to 1 or 0.8 if the gender is wrong. However, the score given is 0.4 which is far from the real translation score and does not give the best information about how well the system is behaving. This problem and many other results are explained in the project at 5.4 Test.

The results of this project are better than expected, the architecture has behaved well despite the lack of input data at training it due to the computational limitations. Analyzing the results it is interesting how the architecture has interpreted the sentences and how behaves with certain uncertainty.

Legally in this project there are few regulations as the data used is of free use. However, there has been a public discuss about the data protection of the sources from some years now. The result of it is the GDPR which makes some rules on how to use this data and the anonymization of it.

The socio-economic impact ,not only by the subject of this project but machine learning in general, is growing year by year. The methods and algorithms are being used in non related working sectors, the only thing needed is data, and data is something that with the past of years become easier to retrieve due to the regulations, improvement and generalization of sensors, data sharing on the internet… so it seems that will continue growing for the coming years.

GLOSARIO DE FIGURAS

1. ÍNDICE DE CONTENIDOS

2. INTRODUCCIÓN

2.1 Motivación del Trabajo

A la hora de introducirme en la temática del aprendizaje profundo para este TFG me ha movido la curiosidad y el interés por el campo del aprendizaje máquina. Desde mi punto de vista, es un campo con un potencial creciente. Cada vez acumulamos más y diversos datos con los que poder trabajar y obtener resultados prometedores. Otro factor a tener en cuenta es el incremento de la potencia computacional a la que tenemos acceso y el desarrollo tecnológico incesante en hardware que permite explotar generalizadamente todos estos datos acumulados.

De un tiempo a ahora se ha visto el impacto que está teniendo este tipo de técnicas relacionadas con el aprendizaje máquina, desde los coches autónomos al tema de este TFG, traducción de lenguaje natural. A muchas personas se les hace imprescindible herramientas que asumimos ya cotidianas en el día a día, como pueden ser los traductores on-line. Viendo la evolución que están tomando las sociedades hacia un mundo más interconectado entre sí, las barreras lingüísticas hacen presencia y herramientas como los traductores se hacen esenciales o útiles como apoyo en las comunicaciones, aprendizaje de idiomas, trabajo… En definitiva, aporta al usuario la capacidad de entender otras lenguas de una manera rápida y sencilla. Situaciones que en otros tiempos podrían ser inaccesibles o tediosas ahora están al alcance de pocos clics.

Otro factor que he tenido en cuenta es el apartado laboral. El campo del big data y todo lo que se relaciona con él posee una gran demanda de recursos humanos actualmente y aparentemente para un futuro. Por ello, trabajar con herramientas y conceptos relacionados con el tema me parecen una buena forma de introducirme en este fascinante mundo desde una perspectiva de ingeniero de telecomunicaciones.

2.2 Objetivos

La idea de la realización de este TFG es tratar de comprender de una mejor forma el modelado de lenguaje natural en el caso particular de traducción de texto. Tratar de diseccionar todos los elementos que componen la arquitectura sequence to sequence y entender qué propósito tiene cada unidad constructiva de la misma, además de los distintos mecanismos y propuestas.

Debido a las limitaciones computacionales no me es posible realizar un traductor relativamente bueno. “We found that the large model configuration typically trains in 2-3 days on 8 GPUs using distributed training in Tensorflow.”[3]. Esta cita muestra la problemática computacional de entrenar este tipo de modelo de una manera completamente óptima. Por ello, he tratado de escalar el problema de forma que pueda obtener un resultado útil, dentro de las posibilidades disponibles.

[3] Denny Britz, Anna Goldie, Minh-Thang Luong, Quoc Le, "Tutorial: Neural Machine Translation"

2.3 Marco regulador

En proyectos relacionados al aprendizaje automático, una de las bases de los mismos son los datos disponibles para su realización. En nuestro caso no hay problemáticas relacionadas ya que usamos datos de uso libre. Sin embargo, como es natural, hay distintas regulaciones en cuanto a qué uso de cuales datos y con qué tipo de consentimiento permiten su explotación. En multitud de aplicaciones de uso diario los usuarios aceptan condiciones de uso que permiten y acreditan a la empresa el almacenaje y explotación de diversos datos. Hay mecanismos de protección del consumidor como el Reglamento General de Protección de Datos (RGPD), el cual instaura preceptos comunes para la comunidad europea como por ejemplo:

"**PRINCIPIO DE RESPONSABILIDAD (ACCOUNTABILITY)**. Habrá que implementar mecanismos que permitan acreditar que se han adoptado todas las medidas necesarias para tratar los datos personales como exige la norma. Es una responsabilidad proactiva. Las organizaciones deben ser capaces de demostrar que cumplen dichas exigencias, lo cual obligará a desarrollas políticas, procedimientos, controles, etc.

PRINCIPIOS DE PROTECCIÓN DE DATOS POR DEFECTO Y DESDE EL DISEÑO. Se deberán adoptar medidas que garanticen el cumplimiento de la norma desde el mismo momento en que se diseñe una empresa, producto, servicio o actividad que implique tratamiento de dato, como regla y desde el origen.

PRINCIPIO DE TRANSPARENCIA. Los avisos legales y políticas de privacidad deberán ser más simples e inteligibles, facilitando su comprensión, además de más completos. Incluso se prevé que, con el fin de informar sobre el tratamiento de los datos, puedan utilizarse iconos normalizados." [4]

En el caso de error en el traductor automático se han producido incidentes, que más que producir acciones legales simplemente se han quedado en sucesos anecdóticos y hasta se podría decir que de chanza. Como es el caso del pueblo de Pontes en Galicia "…their website advertising the local grelo festival. Grelo is a type of vegetable. Unfortunately for them, when Google translated the text from Galician to Spanish, grelo became clitoris." " As Pontes was planning to sue Google over the poor translation (no joke!), but hopefully the town government realized that the money they'd spend on a lawsuit could be better used on trained, human translators and thought better of it."[5]

2.4 Situación actual

El modelado de lenguaje natural se encuentra en un estado de expansión tanto técnica como implementativa. Ya no nos resulta extraño entrar a traductores on-line como pueden ser el de Google o el de DeepL cuando no entendemos cualquier tipo de texto en cualquier tipo de idioma. Tampoco nos resulta extraño la aparición cada vez más frecuente de ChatBots o asistentes on-line, sustituyendo a los tele operadores en casos de atención al cliente que suele ser rutinaria y susceptibles en muchos casos de ser automatizada.

[4] "Reglamento General de Protección de Datos"

[5] Alison Kroulek "Google Translate Mistakes: 6 Times Google Went Rogue"

Claro ejemplo de este tipo de tecnologías son los asistentes por reconocimiento de voz como pueden ser Alexa, Siri… disponibles para el público general, de ventas masivas y de gran utilidad que empiezan a hacer transformaciones sociales y evidenciando la presencia de inteligencias artificiales en nuestro día a día.

En el caso de la traducción automática de texto se suelen utilizar estructuras codificador-decodificador, de las que podemos decir que son relativamente jóvenes. La mayoría de estudios, desarrollos y trabajos de los que vemos sus frutos sobre el tema se condensan en la última década. Además, el crecimiento del sector en todos los sentidos aporta cada día nuevas técnicas y herramientas que permiten ir puliendo los modelos, haciendo que esté en constante evolución y optimización.

Uno de los inconvenientes de este tipo de tecnologías es el costo computacional. Ahora mismo tratar de entrenar modelos competitivos no es una tarea disponible para el público general, los problemas se hacen tan pesados matemáticamente que cierran el abanico permitiendo casi exclusivamente a instituciones y empresas como posibles creadores de ciertos productos.

La perspectiva de futuro de los traductores es la traducción a tiempo real, como pueden ser imagen y reconocimiento de voz, ya presentes en la versión móvil de Google Traductor para imagen y los generadores de subtítulos de vídeos de YouTube para voz, ejemplos completamente disponibles al público general. Sin embargo, estas dos tecnologías todavía son muy mejorables, ya que se están dando los primeros pasos y por el costo computacional que suponen.

2.5 Estructura del proyecto

El proyecto ha consistido en cuatro fases fundamentales:

1. **Estudio e investigación**: debido al desconocimiento de la arquitectura y conceptos, esta fase se ha visto fundamental, en la que se ha recopilado información de diversas fuentes con el fin de obtener una mejor visión para abordar el problema.
2. **Desarrollo del software:** gran parte de la realización del proyecto ha consistido en programación en python sobre tensor flow.
3. **Trabajo de aprendizaje máquina:** tiempo utilizado en la validación, entrenamiento y test del sistema, además del tratamiento de los datos.
4. **Redacción de la memoria:** en esta sección del proyecto se ha seguido recopilando información y estudiando con el fin de obtener la mejor redacción posible de la memoria, así como la interpretación de los resultados.

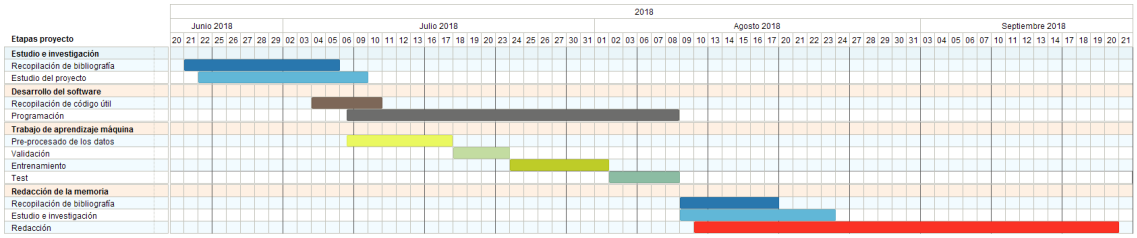

Fig. 2 *Figura detallada en el anexo final.

3. ANÁLISIS DEL PROBLEMA

3.1 Descripción

La resolución de un traductor es un problema complejo, ya que el lenguaje en cada idioma se rige por distintas estructuras gramaticales, pronombres, ausencia de sinónimos literales de uno a otro idioma… todo ello conlleva a que la relación de entrada, en el caso de estudio el inglés, y el idioma de salida, el francés, no sea una búsqueda trivial que se pueda encontrar directamente en una tabla o diccionario.

Uno de los problemas de base que encontramos es la incertidumbre sobre la longitud de la frase a traducir, puede coincidir que la traducción tenga el mismo número de palabras al original pero no suele ser el caso. Añadido a esto, las palabras pueden estar mezcladas o alteradas en orden del original a la traducción, por lo que nuestro sistema, de alguna forma, tiene que aprender el contexto gramatical de cada frase en su idioma. Como ejemplo: “Last night I ate an entire pizza” “Hier soir j'ai mangé une pizza entière”. Podemos observar claramente que las palabras no tienen una correspondencia literal, “Last night” (ultima noche) “Hier soir” (Ayer noche). Además, seguido a eso, nos encontramos con j’ai producto de la liaison gramatical francesa que genera, por así decirlo, la composición o conjunción de dos palabras en una, todo ello proveniente de un sentido fonético del lenguaje que el inglés no posee. Añadido a eso, el francés sí distingue entre femenino y masculino en elementos no personales como ocurre en el castellano y no en inglés, por lo que se produce una relación de la palabra inglesa “the” con numerosas posibles traducciones en francés. Además tiene que atender al contexto de la frase, ya que el femenino y masculino depende de a qué palabra se refiere ese pronombre. Por si fuera poco, en este ejemplo vemos que el orden de aparición de las palabras se ve alterado también: “entire pizza” “pizza entiere”. Como conclusión a todos estos fenómenos vemos una gran dificultad en correlacionar la entrada salida de una forma simple, el sistema deberá aprender estructuras gramaticales y atender al contexto en el que se encuentran las palabras y generan en conjunto.

A parte de los problemas existentes en la lógica de la resolución, la problemática en coste computacional hay que tenerla en cuenta. Partimos del amplio vocabulario que tiene cada idioma, usar todo o una gran parte crea un problema dimensional en los cómputos. Por ello, debemos seleccionar un vocabulario lo suficientemente amplio para representar los dos idiomas pero lo suficientemente ajustado con el fin de hacer unos tiempos computacionales razonables. Entrenar redes neuronales es un proceso lento, si añadimos numerosas dimensiones todo se complica. Acotar las frases a una longitud de palabras límite puede ser una estrategia a tener en cuenta para disminuir dimensiones.

Los problemas computacionales se agravan en mi caso de estudio, ya que los medios disponibles son limitados y realizar un entrenamiento bueno me sería computacionalmente inabarcable, por lo que acotar el problema se me hace inevitable.

3.2 Alcance

Llegar a hacer un traductor razonablemente bueno se torna imposible computacionalmente con mis medios, por lo que obtener resultados estudiables con el fin de poder analizar el comportamiento del sistema y el estudio de la arquitectura son los objetivos a conseguir. Probablemente el sistema se encuentre ante una pobre representación debido al limitado número de muestras computables y el relativamente reducido vocabulario utilizable.

4. ANÁLISIS DE LA ARQUITECTURA

4.1 Sequence to sequence

4.1.1 Arquitectura sequence to sequence

La arquitectura del sequence to sequence se puede caracterizar principalmente por dos elementos constitutivos, codificador y decodificador. En conjunto, la finalidad de esta arquitectura es modelar la probabilidad condicionada p(y|x) de la traducción de una frase de entrada $x_1, \ldots, x_n$ a una frase objetivo $y_1, \ldots, y_n$.

A grandes rasgos funciona de la siguiente forma:

La secuencia de entrada se introduce en una red neuronal recursiva con LSTMs (codificador), a la secuencia objetivo se le añade el símbolo <sos> al inicio de la misma, el cual nos indica el inicio de la frase y nos sirve para almacenar toda la información concerniente a la frase de entrada. Una vez computada la frase de entrada, la información almacenada en el último estado se introduce un una segunda estructura recursiva (decodificador) que irá generando salidas hasta que converja en un símbolo <eos> indicando que ha finalizado la traducción. Cada salida generada en este proceso sirve como entrada del siguiente paso. A la estructura básica se le añaden matrices de embedding y una capa de proyección, cuyas funcionalidades están explicadas en los apartados siguientes (4.2 Embedding).

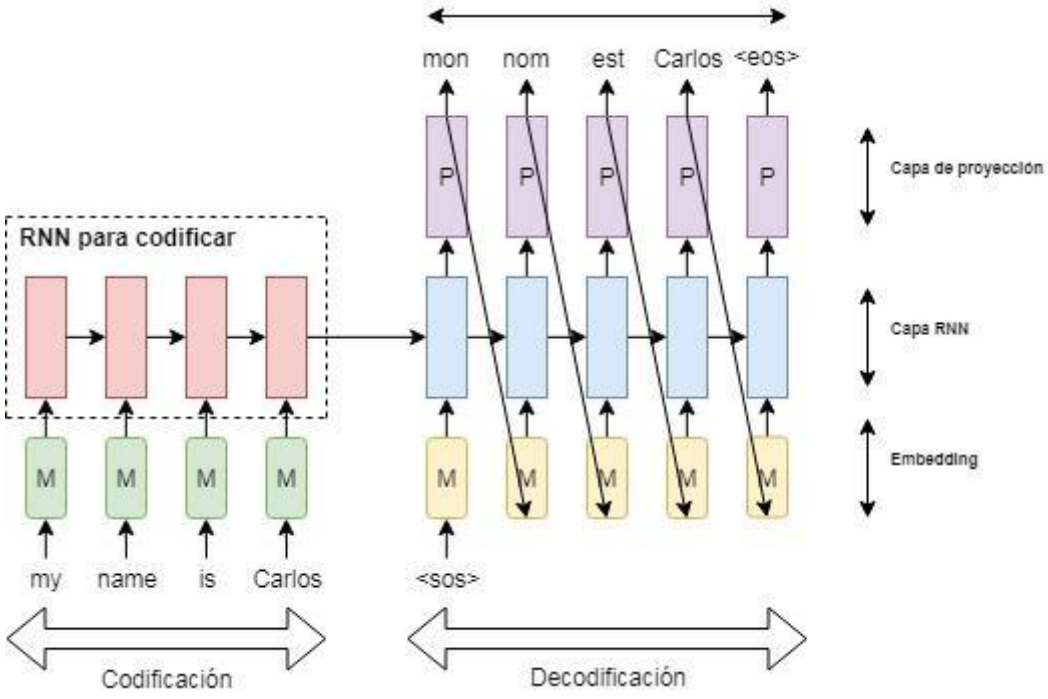

Fig. 1

Lo que realmente está haciendo la red es generar distribuciones de probabilidad sobre el vocabulario. Es la probabilidad de que la palabra de salida sea una palabra del vocabulario dada toda la frase de entrada y las salidas obtenidas hasta ese momento.

$$Y_t = P(O = w|O_{t-1}, ..., O_t, X_1, ..., X_n)$$

De esta forma en cada salida tenemos una distribución de probabilidad sobre el vocabulario del lenguaje en forma vectorial, dada la secuencia de entrada y las salidas obtenidas hasta el momento. De la distribución de probabilidad obtenida se selecciona una o varias palabras (explicadas las técnicas greedy y beam search más adelante) y se utilizan como entradas del siguiente paso. Este proceso se realiza de forma recursiva hasta obtener el símbolo <eos>.

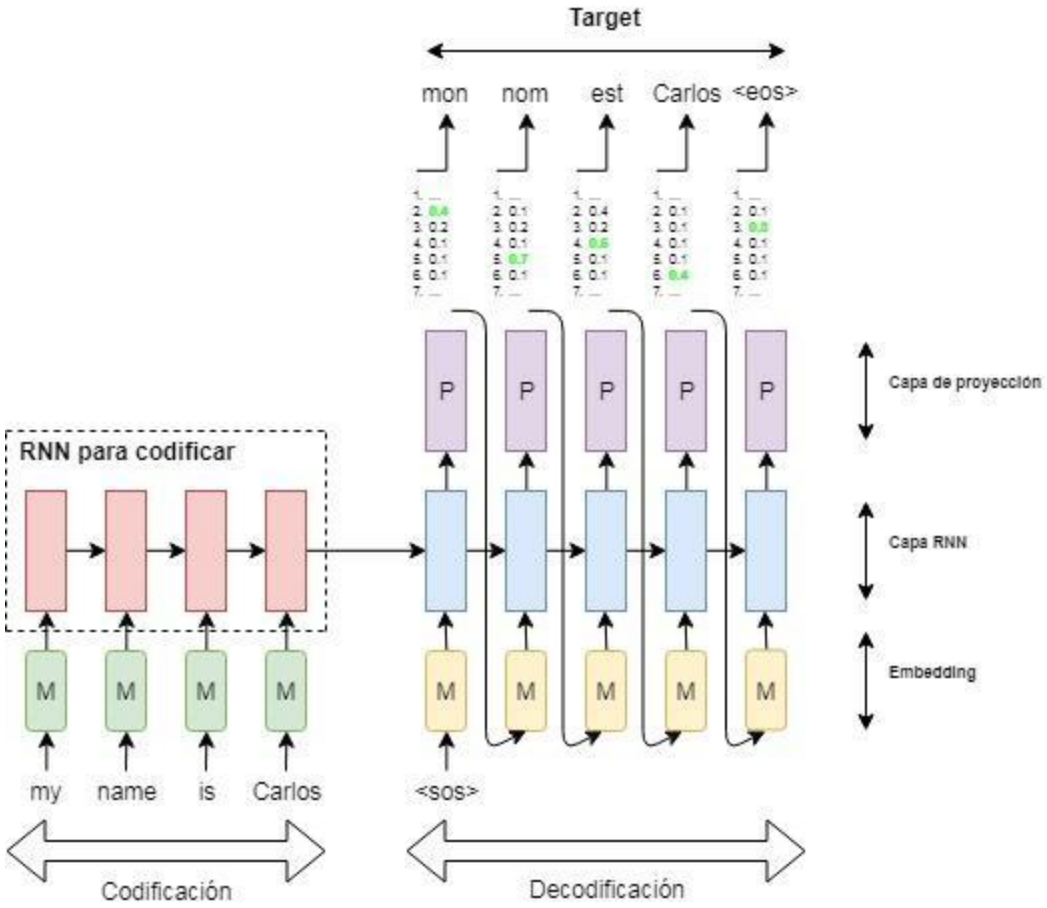

Fig. 3

De la probabilidad de cada palabra seleccionada, podemos obtener la probabilidad de que la frase obtenida en la traducción sea la deseada o más próxima a lo ideal. Resulta tal que la probabilidad de la frase es el productorio de las probabilidades de las salidas (palabras) obtenidas. Por lo que debemos seleccionar las frases con máxima probabilidad que podamos obtener.

$$P(O_1, ..., O_L|X_1, ..., X_L) = y_{O_1} y_{O_2} ... y_{O_n}$$

$$Argmax(y_{O_1} y_{O_2} ... y_{O_n})$$

A la hora de seleccionar las frases óptimas y tratar de maximizar las funciones anteriores para obtener la máxima probabilidad, no enfrentamos al siguiente problema, somos incapaces

de saber a priori qué selección de palabras originará una mayor probabilidad final. Nos vemos limitados a saber la probabilidad futura ya que estamos atados a que posiblemente elegir de las siguientes palabras la que tenga mayor verosimilitud haga que nuestra frase en conjunto acabe degenerando en palabras con una verosimilitud peor degradando la probabilidad final.

Es decir, si en la siguiente salida, constituida por una distribución de probabilidad sobre nuestro vocabulario, obtenemos una distribución parecida a la normal, o varias palabras poseen una probabilidad similar, no tenemos pistas de qué palabra puede llevarnos a una traducción final prometedora. Al hacer el producto de la palabra seleccionada anterior con la posible seleccionada de este momento obtenemos probabilidades similares, genera incertidumbre, y es posible que escojamos mal la palabra y en el paso siguiente la frase deje de tener sentido. Debido a este problema, escoger y probar solo una palabra lleva a peores traducciones ya que el productorio de probabilidades de las salidas hasta ese momento es el máximo, sin embargo, no sabemos la probabilidad de la frase total a la que convergerá a priori. Es posible que seleccionando una palabra que resulte una frase menos probable acabe en las consecutivas iteraciones generando una frase más prometedora.

Para la resolución de nuestro problema es más interesante algoritmos de ramificación como puede ser beam search. El algoritmo se basa en que dada nuestra limitación a la hora de saber a priori qué combinación de palabras final puede ser la más prometedora, realiza una ramificación seleccionando las n mejores frases, y de ellas se queda con la más prometedora. En nuestra arquitectura funciona de la siguiente forma: suponiendo que hacemos un beam search seleccionando las 2 mejores, n = 2, el algoritmo se ejecuta de la forma que de cada salida en cada momento, selecciona las 2 mejores posibles palabras y las utiliza como entrada en la siguiente iteración de forma recursiva. Así finalmente tendremos en este caso 2 posibles frases haciendo nuestro sistema más robusto frente a la mala selección de palabras.

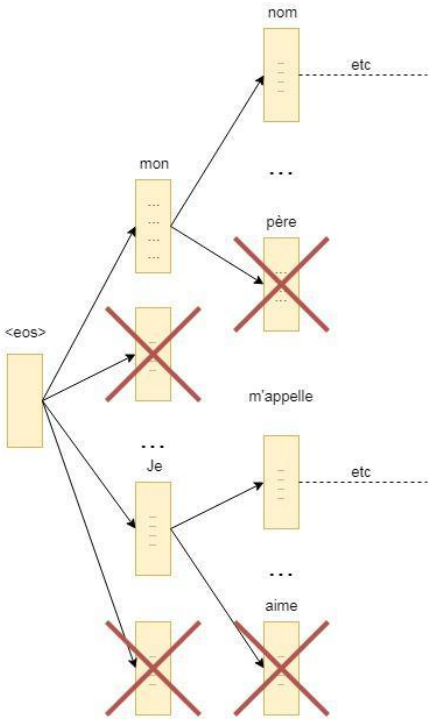

Fig. 4

4.1.2 Entrenamiento

A la hora de entrenar la arquitectura se entrena en conjunto, es decir, se entrenan tanto las matrices de embedding (una distinta para cada idioma) y las redes neuronales recursivas compuestas por celdas LSTM. Para entrenar nuestro sistema necesitamos frases de un idioma y otro perfectamente traducidas y emparejadas una a la otra, de esta forma en el entrenamiento la entrada del codificador serán las frases del idioma original y en el decodificador trabajarán como entradas las frases en el idioma a traducir. El proceso se hace de forma secuencial, de forma que vamos entrenando pareja a pareja. La salida es una distribución de probabilidad sobre el vocabulario objetivo. Se computa la divergencia entre la salida y las palabras objetivo y se refrescan los pesos de la red propagando las derivadas de la divergencia.

Según : "…LSTM learns much better when the source sentences are reversed (the target sentences are not reversed)." Por lo que las entradas al codificador se realizan de forma inversa, alimentando primero con la última palabra, después la penúltima… consecutivamente.

"While we do not have a complete explanation to this phenomenon, we believe that it is caused by the introduction of many short term dependencies to the dataset. Normally, when we concatenate a source sentence with a target sentence, each word in the source sentence is far from its corresponding word in the target sentence. As a result, the problem has a large "minimal time lag" [6]. By reversing the words in the source sentence, the average distance between corresponding words in the source and target language is unchanged. However, the first few words in the source language are now very close to the first few words in the target language, so the problem's minimal time lag is greatly reduced. Thus, backpropagation has an easier time "establishing communication" between the source sentence and the target sentence, which in turn results in substantially improved overall performance."[7]

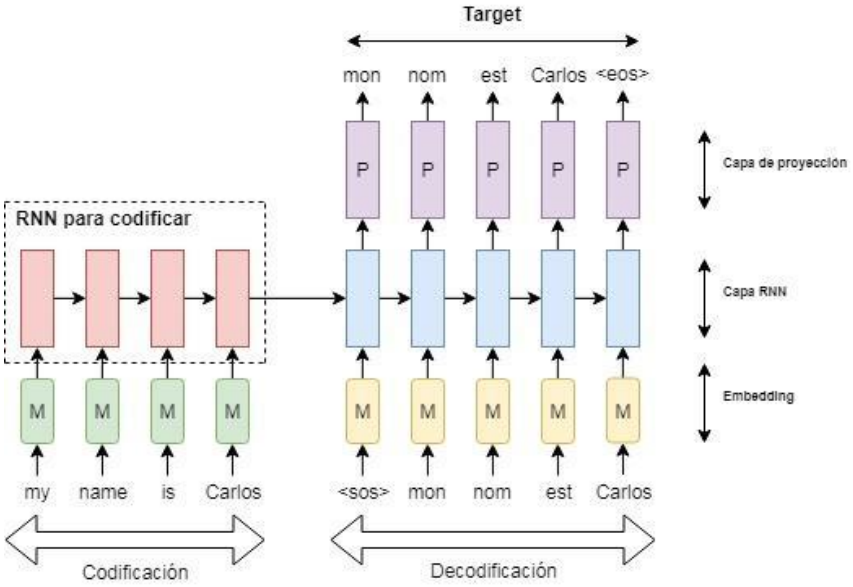

Fig. 5 *En la figura no se muestra en la entrada del codificador la frase invertida, en la práctica como indicado anteriormente si está invertida.

[6] S. Hochreiter and J. Schmidhuber. "Sequence to Sequence Learning with Neural Networks." 1997.
[7] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. "LSTM can solve hard long time lag problems." 2014.

4.2 Embedding

Las técnicas de embedding son algoritmos de aprendizaje de estructuras de espacio euclídeo para palabras. Las arquitecturas pueden variar desde redes neuronales, redes convolucionales, o redes neuronales recursivas el cual es nuestro caso.

Realizar embedding nos permite transformar nuestras frases en vectores de números reales de una dimensión menor que el vocabulario que utilizamos en los dos idiomas. Esto quiere decir que para representar cada palabra, no necesitamos asignarle un valor único o índice en un vector del tamaño del vocabulario, como podría ser haciendo one-hot encoding, haciendo embedding podemos asignar a cada palabra un valor vectorial menor al tamaño del vocabulario original. De esta forma el costo computacional de las operaciones en las posteriores secciones de la arquitectura se ve disminuido.

El factor más importante a la hora de tener en cuenta el uso de este tipo de técnicas es que transforma el valor discreto que nosotros le podamos dar a un valor continuo. Al ser un valor continuo permite su uso para realizar back propagation en su aprendizaje. La lógica básica de aprendizaje es entrenar nuestra red neuronal para que dando una frase de entrada, y seleccionando una palabra dentro de una ventana en esa frase, obtener las probabilidades de palabras adyacentes pertenecientes a nuestro vocabulario en forma de parejas.

the	prime	minister	is	speaking

En la figura anterior podemos observar la frase the prime minister is speaking, en este caso la ventana seleccionada es de tamaño 2 para facilitar la comprensión. Dentro de la frase se ha elegido aleatoriamente la palabra prime, de ella obtenemos las parejas (prime, the), (prime, minister) , (prime, is) , (prime, speaking). Así, la red neuronal entrenando va a ir aprendiendo las estadísticas relacionadas con las palabras y cómo están relacionadas entre ellas. Dentro de la capa oculta, no hay función de activación pero en la capa de salida si se usa softmax. El resultado final que queremos obtener reside en la capa oculta de la red neuronal, los pesos de esa capa oculta son los valores buscados.

Cabe señalar que utilizamos un embedding distinto tanto para la parte de codificación y decodificación, de esta forma entrenaremos para inglés una matriz de embedding distinta e independiente de la matriz de embedding de francés. Además, estos embeddings se van entrenando a la vez que el resto del modelo.

Un embedding bien entrenado puede tener múltiples funciones, ya que las palabras se agrupan por significados parecidos o relaciones entre ellas haciendo posible técnicas de clustering. Añadido a eso se pueden hacer posibles operaciones entre los valores vectoriales de las mismas aprovechando sus relaciones. Por ejemplo, en un hipotético caso, si hiciésemos la siguiente operación vectorial: rey - hombre + mujer, nos resultaría reina o un valor aproximado a ella.

Además realizar embeddings en distintos idiomas resultan en caracterizaciones dentro de este parecidas, realizando un PCA sobre el embedding para poder visualizar la disposición en el plano de las palabras nos permite entenderlo mejor.

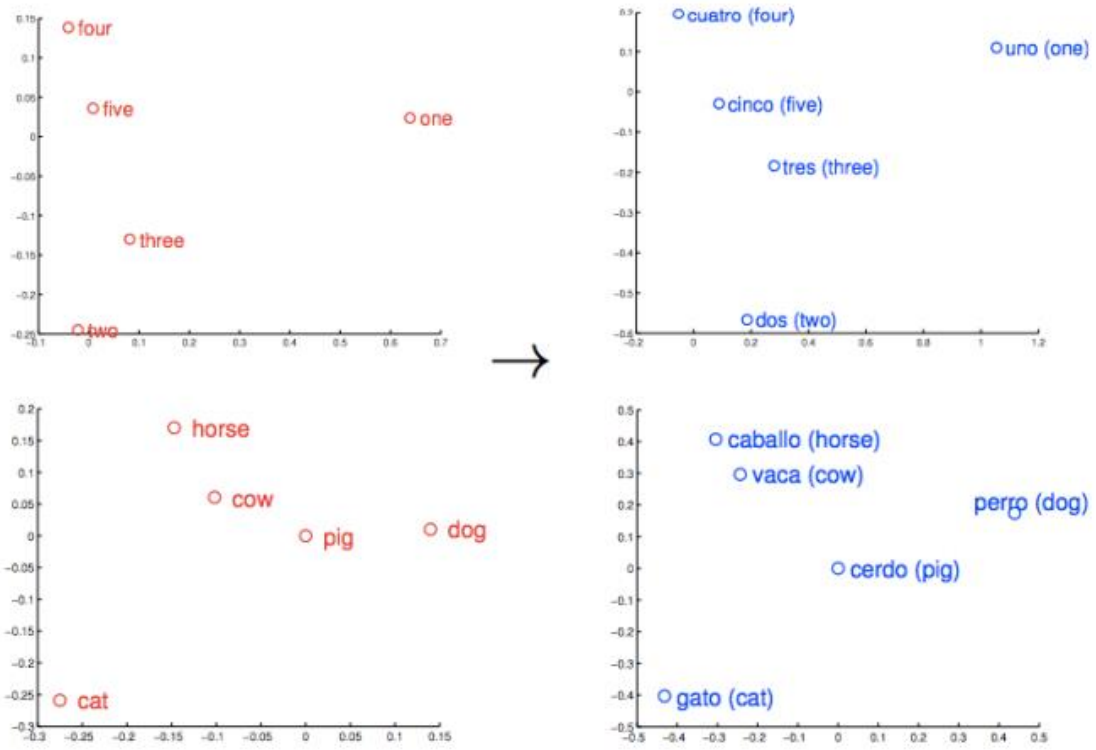

Fig. 6[8]

Como se puede ver, las palabras se disponen y relacionan entre ellas de similar forma, aunque no exactamente igual, a pesar de ser dos embeddings e idiomas distintos.

Para recuperar la expresión original a partir del embedding partimos de una dimensión menor en una matriz densa, por lo que usamos la denominada capa de proyección. Esta capa nos proporciona transformar las salidas en dimensiones relacionadas con el embedding a las dimensiones originales para de esta forma obtener las palabras finales y evaluarlas. Es decir, al estar computando nuestro modelo todas las operaciones en las dimensiones del embedding, para poder obtener la salida real final hay que ampliar esas dimensiones a las de nuestro vocabulario original. Fundamentalmente necesitamos tener el embedding y la capa de proyección por separado porque la función de pérdida necesita acceso simultáneo a las dos capas.

[8] "Word2Vec, Doc2vec & GloVe: Neural Word Embeddings for Natural Language Processing"

4.3 LSTM

LSTM (Long short-term memory) presentadas por Sepp Hochreiter y J□urgen Schmidhuber [9] son las unidades estructurales dentro de nuestra red neuronal recursiva. Lo que nos permite, a grandes rasgos, es continuar el aprendizaje con un contexto previo. Conocer el contexto previo y no aprender cada elemento desde cero se torna fundamental a la hora de comprender el lenguaje natural.

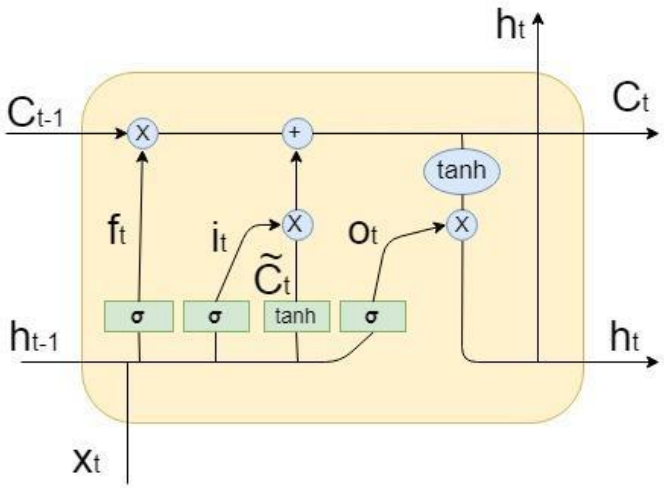

Fig. 7

El funcionamiento de la LSTM atendiendo a la figura anterior es el siguiente:

1) Seleccionar qué parte queremos olvidar, de ello se encarga la primera sigmoide de la izquierda. Forma la capa forget. Tomará h_{t-1} y x_t y saca un valor entre 0 y 1, siendo 0 completamente olvidado y 1 lo deja intacto.

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f)$$

2) En el siguiente paso seleccionamos la parte que queremos recordar. De ello se encarga la segunda sigmoide seleccionando los valores a recordar o actualizar y la tanh crea un vector de posibles candidatos a añadir.

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i)$$
$$\widetilde{C}_t = \sigma(W_c[x_t, h_{t-1}] + b_c)$$

3) Multiplicamos nuestra capa de olvidar o forget con C_{t-1} y añadimos nuestra capa a recordar obtenida en el segundo paso. El resultado es C_t.

$$C_t = \sigma(f_t C_{t-1} + i_t \widetilde{C}_t)$$

4) El paso final actúa de filtro y selecciona la parte que queremos que sea nuestra salida. La tanh nos permite forzar que los valores sean entre -1 y 1. La salida del paso 3 C_t entra en la celda de la tanh y la multiplicamos por la sigmoide final. Obteniendo de esta forma h_t.

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \qquad h_t = tanh(C_t)o_t$$

[9] S. Hochreiter and J. Schmidhuber. “long short-term memory.” 1997.

5. DISEÑO DEL MODELO Y DESARROLLO DEL PROBLEMA

5.1 Pre procesado del texto

Antes de entrar en el pre procesado mismo del texto quiero hacer mención a la fuente del mismo. El dataset elegido es una recopilación de intervenciones en el europarlamento [10] cabe destacar que el lenguaje empleado en un parlamento suele distar de las expresiones y el lenguaje utilizado coloquialmente. Además posee múltiples referencias a nombres, regiones, países… por lo que podemos considerar que es un texto en conjunto rico en vocabulario aunque puede que difiera del uso coloquial de los dos idiomas.

Para elegir el idioma de traducción he tenido en consideración la similitud entre idiomas en lo referente al vocabulario, si los idiomas distan mucho entre sí es posible que se generen muchas frases con poca información o dispersa. La siguiente gráfica muestra la relación del entrenamiento del idioma inglés con distintos lenguajes teniendo en cuenta el número de palabras que difieren en cada uno.

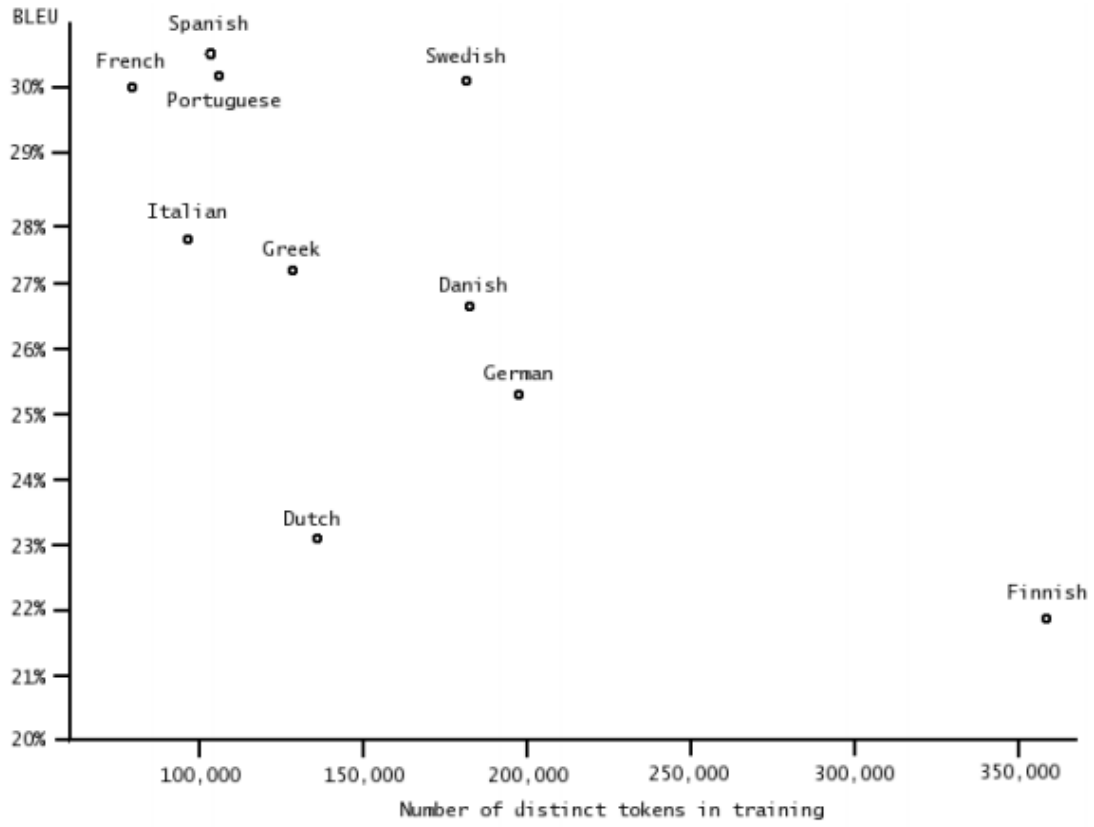

Fig. 8[11]

[10] "European Parliament Proceedings Parallel Corpus 1996-2011"

[11] "Europarl: A Parallel Corpus for Statistical Machine Translation"

Debido a estas características me he decantado por el uso del francés, ya que posee una cierta similitud con el inglés y puede facilitar la resolución del problema. Como características del dataset original tenemos 2,007,723 frases relacionadas una a una siendo su traducción de un idioma al otro. De las frases, tenemos 51,388,643 palabras en francés y 50,196,035 en inglés.

Para poder trabajar con el dataset hay que hacer una serie de procesos con el fin de normalizar nuestro texto. Para ello he convertido todo el texto a minúscula, dentro del texto en francés reside el problema que utiliza tildes y letras que no posee el inglés, por lo que he decidido truncar el problema transformando las letras a un equivalente común (las letras que posee el inglés) y eliminando todas las tildes. Seguido a eso he eliminado todos los caracteres que no son imprimibles y también las palabras o partes con elementos numéricos he decidido suprimirlas. Con el resultado de todos estos procesos ha sido el elemento básico con el que empezar a trabajar.

Después de adecuar el texto de forma que tenga unas características básicas de normalizado, los pasos siguientes han estado relacionados con acotar y seleccionar dentro del texto las características o propiedades que puedan ser de interés en la ejecución del problema. En primer lugar, hay que definir un vocabulario que componga de la mejor forma posible el texto disponible. Dentro de esta definición de vocabulario hay que tener en cuenta las limitaciones computacionales y que cantidad de palabras pueden ser suficientemente buenas para comprender el vocabulario, de todas las palabras cuales son las más óptimas y que hacer con las palabras que "olvidamos"

Para elegir el número total de palabras de vocabulario me he regido por publicaciones de artículos académicos y divulgativos relacionados con la cantidad de palabras con las que se puede tener cierta completitud o dominio de un idioma[12] [13].

Decantándome finalmente por la cantidad aproximada de 10,000 palabras "Non-native English users generally reach 10,000+ words by living abroad" [14]. Disponiendo de una cantidad a la que aproximarse falta discernir las palabras que definirán ese vocabulario, para ello dentro de cada uno de los dos textos he conteado las palabras según su número de veces de aparición dentro del mismo. Teniendo estos datos he seleccionado las que se han repetido cierto número de veces o más hasta completar un número cercano a 10,000. En concreto, el vocabulario final es de 10,152 palabras para inglés y 10,115 para el francés. Una vez sabido las palabras que componen mi vocabulario falta decidir qué hacer con las palabras a olvidar o no saber ya que no están definidas dentro de mi vocabulario. La decisión final ha sido sustituir todas las palabras no comprendidas en el vocabulario por una palabra conocida la cual he decidido nombrar unk proveniente de unknown (desconocido). Al vocabulario seleccionado de cada idioma he añadido la palabra unk y distintos tokens o elementos como el <eos> y <sos> utilizados en la lógica de la arquitectura sequence to sequence.

[12] "¿Cuántas palabras se necesitan para hablar con fluidez un idioma?" Course Finder
[13] "¿Cuántas palabras se necesitan para comunicarse?" BBC
[14] "How many words are in the average English speaker's working vocabulary?" Robert Charles Lee

Ejemplo de los textos una vez preprocesados:

- INGLÉS

resumption of the session
i declare resumed the session of the european parliament adjourned on friday december and i would like once again to wish you a happy new year in the hope that you enjoyed a pleasant unk period

although as you will have seen the unk millennium unk failed to materialise still the people in a number of countries suffered a series of natural disasters that truly were dreadful

you have requested a debate on this subject in the course of the next few days during this partsession
in the meantime i should like to observe a minute s silence as a number of members have requested on behalf of all the victims concerned particularly those of the terrible storms in the various countries of the european union
please rise then for this minute s silence
the house rose and observed a minute s silence
madam president on a point of order
you will be aware from the press and television that there have been a number of bomb unk and killings in sri lanka
one of the people assassinated very recently in sri lanka was mr unk unk who had visited the european parliament just a few months ago

- FRANCÉS

reprise de la session
je declare reprise la session du parlement europeen qui avait ete interrompue le vendredi decembre dernier et je vous unk tous mes vux en esperant que vous avez passe de bonnes vacances
comme vous avez pu le constater le grand unk de lan ne sest pas produit en revanche les citoyens dun certain nombre de nos pays ont ete victimes de catastrophes naturelles qui ont vraiment ete terribles
vous avez souhaite un debat a ce sujet dans les prochains jours au cours de cette periode de session
en attendant je souhaiterais comme un certain nombre de collegues me lont demande que nous unk une minute de silence pour toutes les victimes des tempetes notamment dans les differents pays de lunion europeenne qui ont ete touches
je vous invite a vous lever pour cette minute de silence
le parlement debout observe une minute de silence
madame la presidente cest une motion de procedure
vous avez probablement appris par la presse et par la television que plusieurs attentats a la bombe et crimes ont ete perpetres au sri lanka

lune des personnes qui vient detre unk au sri lanka est m unk unk qui avait rendu visite au parlement europeen il y a quelques mois a peine

Como se puede apreciar ya en los ejemplos anteriores hay una fuerte presencia de frases largas y compuestas. Para simplificar la tarea computacionalmente he decidido truncar y quedarme con las frases que estén comprendidas por 10 o menos palabras, trabajar con más palabras aumenta las dimensiones computacionalmente hablando, imposibilitando su resolución con los medios disponibles.

5.2 Validación

Para la validación no he tenido que hacer una elección de parámetros concienzuda. Me he limitado a probar mediante una validación cruzada con parámetros que me ofreciesen una flexibilidad tal del modelo que me permitiese converger a valores de pérdida bajos en un tiempo razonable. Posiblemente mi sistema se encuentre sobre ajustado en cierta medida, sin embargo con las limitaciones computacionales que dispongo realizar un traductor óptimo es una tarea irreal. Además, suavizar el entrenamiento como previsión de cualquier sobre ajuste inflexibilizando el sistema no ha aportado mejoras en el test final, si no una degradación considerable en el rendimiento. El camino por el que he optado trabajar creemos es el más propicio para estudiar la arquitectura dentro de las restricciones impuestas.

Para la validación he utilizado un 20% de las frases totales con las que trabajo como dataset, quedando un total de 6,968 frases. Finalmente los parámetros de la arquitectura y de la lógica han resultado tal que:

Parámetro	**Valor**
Tamaño del batch	400
Número de unidades dentro del LSTM	512
Tamaño del embedding	1000
Learning rate	0.001
Max gradient norm	5.0
Beam width (beam search)	3
Épocas	150

Para tratar de obtener una caracterización del sistema se ha tenido en cuenta la pérdida atendiendo a la función de *tensorflow sparse_softmax_cross_entropy_with_logits*[15]. Tiendo un desarrollo de la misma durante las épocas de esta forma:

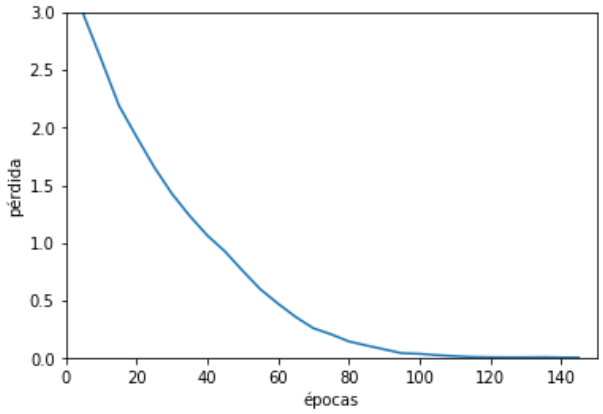

Fig. 9

A la hora de computar un score he realizado una comparación de la frase resultante con la objetivo, se compara palabra a palabra en su respectiva posición, finalmente se obtiene un porcentaje de palabras coincidentes entre la resultante y objetivo. El score obtenido en la validación de esta forma es tal que 0.97989449113 para una decodificación *greedy*, y 0.979655520542 utilizando una decodificación *beam search*. Cabe destacar la prácticamente insignificante degradación usando *beam search,* pero es posible que sea debido al sobreajuste.

Recuerdo que el objetivo no es realizar un traductor óptimo si no obtener resultados estudiables con el fin de comprender el comportamiento de la arquitectura.

5.3 Entrenamiento

Para el entrenamiento, el número de frases utilizadas son un total de 20,908. Los parámetros de ajuste de la arquitectura son los obtenidos en la etapa de validación a excepción de las épocas. Del entrenamiento resulta una evolución de la pérdida frente a las épocas con la forma siguiente:

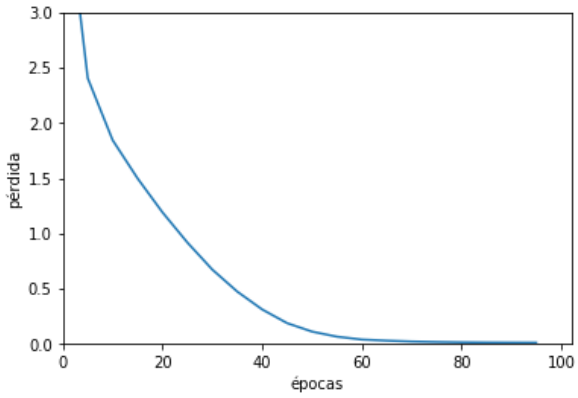

Fig. 10

[15] "tf.nn.sparse_softmax_cross_entropy_with_logits" TensorFlow

Produciéndose como score 0.965996265173 con una decodificación greedy, 0.966440242764 con una decodificación beam search. El sistema diseñado ofrece la flexibilidad que necesitamos en primera instancia para obtener unas pérdidas bajas para el número de frases utilizado en el entrenamiento. Además en este caso beam search, como esperado, mejora al método greedy.

5.4 Test

Los resultados en el test son notablemente buenos para el tamaño del dataset utilizado en el entrenamiento, produciéndose como resultado 0.243138282061 para la decodificación greedy y 0.245294918725 para la decodificación beam search. Significando, que para ambos casos las frases son en promedio ¼ exactamente iguales a las frases objetivo. Además, los resultados obtenidos son con una evaluación pesimista, ya que no contempla la utilización de sinónimos o la construcción alternativa de oraciones resultando en un significado igual o similar. El método de evaluación simplemente contempla la aparición de las palabras en el mismo orden y de forma exacta. Ilustrar con ejemplos ayuda en la apreciación de lo referente a la evaluación.

Los siguientes ejemplos muestran en primer lugar la frase en inglés la cual queremos traducir, en segundo lugar la frase en francés obtenida mediante nuestro traductor, en tercer lugar la frase a la que debe aspirar, objetivo (target) el traductor. Finalmente se representa el score evaluado en esa frase.

Las frases están representadas en los términos del preprocesado, sin tildes, en minúscula…

Entrada	thank you very much commissioner
Salida	merci beaucoup monsieur le commissaire
Objetivo	merci infiniment madame la commissaire
Score	0.4

Analizando el ejemplo anterior podemos apreciar mejor la problemática en la evaluación ya que representa un resultado pesimista. La salida del traductor utiliza beaucoup en vez de infiniment, que pueden ser relativamente sinónimos variando la cantidad. El restante de la frase confunde el género y le asigna el masculino en vez del femenino. Es un error comprensible ya que la frase en inglés no especifica el género y al traducir en francés forzosamente tienes que asumir el género de le/la commissaire. Analizando la traducción el score humano seguramente sería más que 0.4, pero atendiendo a cómo está computado el score lo podemos entender como un índice de similitud mínimo sin embargo con una proyección más optimista que la que aporta.

Entrada	in fact it is doing the opposite
Salida	en fait cest exactement linverse
Objetivo	bien au contraire
Score	0.0

En este caso es aún peor, ya que estamos aportando un score nulo a una frase que se podría considerar traducción literal de la entrada en inglés. Podríamos decir que es una frase correctamente traducida pero la computamos como completamente errónea.

Más ejemplos variados:

Entrada	thank you commissioner liikanen
Salida	je vous remercie monsieur le commissaire liikanen
Objetivo	merci beaucoup Monsieur le commissaire liikanen
Score	0.0

Entrada	the vote will take place tomorrow at noon
Salida	le vote aura lieu demain a heures
Objetivo	le vote aura lieu demain a heures
Score	1.0

Entrada	the formal sitting was closed at pm
Salida	la seance solennelle est close a
Objetivo	la seance solenenelle est levee a
Score	0.833

Entrada	this is what I have to soy to you
Salida	voila ce que je voulais vous dire
Objetivo	je dois vous unk la chose suivante
Score	0.0

Entrada	the original intention was to phase out these measures
Salida	la liste elaboree en annexe convient a bien sur ce point
Objetivo	au depart il safissait de supprimer progressivement ces mesures
Score	0.0

Entrada	all of these things have been very beneficial since
Salida	tous ces objectifs ont ete annule
Objetivo	toutes ces choses ont ete tres benefiques depuis
Score	0.375

Entrada	that is important especially in the area of tax
Salida	cest partiulierement important au sein de lunion
Objetivo	cest important en particulier dans le domaine
Score	0.125

Entrada	we shall therefore vote against this amendment
Salida	nous voteron contre contre le rapport
Objetivo	nous voteron donc contre cet amendement
Score	0.5

Entrada	the commission cannot accept it
Salida	la commission ne peut accepter les choses
Objetivo	la commission ne peut laccepter
Score	0.5713

6. ENTORNO SOCIO-ECONÓMICO Y PRESUPUESTO

La herramienta desarrollada en este TFG de una forma óptima, ha realizado y realizará transformaciones sociales de forma directa y global. La mejora y abaratamiento del coste computacional de forma continua hace posible la facilidad de acceso a todo este tipo de instrumentos y cada vez más complejos. En el caso de estudio, produce sin lugar a dudas un acercamiento entre gentes y pueblos, ya que facilita el entendimiento interlingüístico derribando numerosas barreras a las que otrora sólo podían aspirar a tumbar intérpretes o traductores. Además, en el apartado académico, puede ser una herramienta muy útil para el aprendizaje de idiomas, teniendo acceso a traducciones de forma rápida que después pueden ser consultadas con conocedores del idioma.

En el entorno empresarial, poder realizar traducciones de los productos o dentro del intercambio comercial ayuda a la globalización y a la flexibilidad en las importaciones y exportaciones, poder comunicarse entre los interesados directamente minimizando la intervención de intermediarios abarata los costes definitivamente.

En el sector laboral que atañe el proyecto tiene una incisión clara, ya que puede parecer que reemplaza la figura del traductor o intérprete. En cierta medida es cierto que la herramienta invade ciertos nichos antes acaparados totalmente por traductores, sin embargo, los idiomas son difíciles de mecanizar. La problemática en la traducción si se empieza a complejizar el texto es que la comunicación contiene contenido entre líneas, contextos, localismos… que de una forma automática es difícil de captar e interpretar y en la que la figura de los traductores se torna imprescindible para entender la completitud de lo que se quiere decir.

Proyectando esta tecnología y las relacionadas con la inteligencia artificial hacia el futuro, son susceptibles de conformarse como sustitutivos de numerosos empleos y sectores o como herramientas de apoyo. Seguramente, como en todo avance o revolución aparecerán grupos de afectados o gente descontenta. Realmente es un elemento cíclico en la historia, el enfrentamiento del hombre contra la máquina, en el que el hombre desde mi punto de vista se verá siempre abocado al “fracaso”. Cambiar los animales de arrastre o tiro por motores, cambiar las tejedoras por máquinas automatizadas… son alteraciones sociales importantes, pero ya nadie se imaginaría yendo al trabajo o de vacaciones en calesa.

El tfg se toma como un primer contacto e inicio de desarrollo de la tecnología, el presupuesto se proyecta para el próximo año. El proyecto requeriría de dos trabajadores, el investigador y el graduado. En términos de costes:

El Grupo de Tratamiento de la Señal y Aprendizaje (GTSA) proporcionaría los medios computacionales necesarios para el desarrollo de la tecnología. A parte, se precisaría de un ordenador personal para el graduado. El software utilizado y los datos utilizados son libres por lo que no será necesario comprar licencias.

Añadido a esto, se suman los salarios del graduado a jornada completa, y del investigador. El salario del graduado se acoge a las tabulaciones oficiales de la UC3M [16]. El investigador (tutor) sería remunerado por la universidad.

SUELDO		
Grupo	mensual	anual (15 pagas)
A1	2.273,78	34.106,74
A2	2.102,31	31.534,71
B1	1.905,79	28.586,87
B2	1.829,00	27.434,98
C1	1.639,67	24.595,02
C2	1.575,60	23.633,93
C3	1.411,44	21.171,61
D	1.316,08	19.741,23

Fig. 11

Siendo el monto total del coste salarial 31.534,71€

Elemento	**Coste**
Ordenador personal	1000€
Congresos y viajes	2000€
Salario	31.534,71€
Total	34.534,71€

16

https://www.uc3m.es/ss/Satellite/RHPas/es/Detalle/Ficha_C/1371246114151/1371245242201/Tablas_Retributivas

7. CONCLUSIONES

7.1 Objetivos cumplidos

Los objetivos propuestos, en nuestro caso el análisis y estudio de la arquitectura sequence to sequence ha sido satisfactorio. Hemos obtenido resultados interesantes a pesar de usar un dataset reducido comparado con el utilizado para proyectos óptimos de este tipo. Las traducciones obtenidas han sido prometedoras, mostrando cómo el sistema busca sinónimos, o formas distintas de llegar a la misma traducción que los traductores humanos. Cabe destacar las limitaciones de la estructura a la hora de traducir frases de significante longitud, a mayor longitud el sistema tiende a degradarse en su rendimiento.

7.2 Mejoras

La arquitectura estudiada en este trabajo de final de grado posee ciertas limitaciones que han sido subsanadas en gran medida con la implementación de nuevas arquitecturas. La más reseñable e influyente en el campo de la traducción con deep learning es attention, el cual trata de contextualizar las frases de entrada facilitando la tarea del codificador. La arquitectura básica codificador-decodificador trata de condensar toda la información concerniente a la frase de entrada en un vector fijo, ésto resulta en un cuello de botella sobretodo para frases de mayor longitud.

La lógica que reside en attention es la de crear anotaciones de la frase de entrada, para ello utiliza dos redes neuronales recursivas (RNN) bidireccionales. Se necesita una bidireccionalidad entre otros motivos porque las RNN tienden a aprender mejor los primeros pasos de las frases degradándose en los últimos. De esta forma, guarda una anotación en cada palabra de la frase conteniendo información no solo de la palabras precedentes, también en las consecutivas. Concatenando la información de las precedentes y de las consecutivas tenemos información de las palabras que rodean la palabra en estudio.

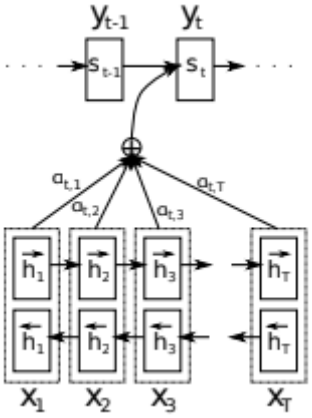

Fig. 12[17]

El vector de contexto (Fig. 12) es una suma de las anotaciones explicadas en el apartado anterior, donde α (Fig. 13) representa un peso que caracteriza la importancia de cada

[17] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio "Neural Machine Translation by Jointly Learning to Align and Translate"

anotación y sus relativos para la obtención del siguiente estado en la traducción. Para obtener α se utiliza un sistema de alineación que representa lo bien que encajan las entradas en j y las salidas en i.

Este sistema mejora al anterior consiguiendo aliviantar la tarea del codificador, haciendo un sistema más adaptativo y dinámico. Los resultados obtenidos con attention en distintas áreas y proyectos señalan cierta mejoría.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

Fig. 13[18]

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

Fig. 14[19]

[18] Ibid

[19] Ibid

BIBLIOGRAFÍA

CMU Deep Learning “S18 Sequence to sequence models: Attention Models” https://www.youtube.com/watch?v=oiNFCbD_4Tk

Word2Vec Tutorial - The Skip-Gram Model http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

https://deeplearning4j.org/word2vec.html

Word2Vec Tutorial Part 2 - Negative Sampling http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

Understanding LSTM Networks http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM and GRU -- Formula Summary https://isaacchanghau.github.io/post/lstm-gru-formula/

Neural Machine Translation (seq2seq) Tutorial https://github.com/tensorflow/nmt#wmt-german-english

Thang Luong’s Thesis on Neural Machine Translation https://github.com/lmthang/thesis

Seq2Seq model in TensorFlow https://towardsdatascience.com/seq2seq-model-in-tensorflow-ec0c557e560f

Mihn-T. Luong, Hieu Pham, Christopher D. Manning “Effective Approaches to Attention-based Neural Machine Translation

Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio “Neural Machine Translation by Jointly Learning to Align and Translate”

Grégoire Mesnil, Xiaodong He, Li Deng, Yoshua Bengio. “Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding”

Ilya Sutskever, Oriol Vinyals, Wuoc V.Le “Sequence to Sequence Learning with Neural Networks”

Denny Britz, Anna Goldie, Minh-Thang Luong, Quoc Le, "Tutorial: Neural Machine Translation", [En línea]. Disponible en: https://google.github.io/seq2seq/nmt/.

“Reglamento General de Protección de Datos” [En línea] Disponible en: https://rgpd.es/

Alison Kroulek “Google Translate Mistakes: 6 Times Google Went Rogue” 3/2/2016 [En línea] Disponible en: https://www.k-international.com/blog/google-translate-mistakes/

Ilya Sutskever, Oriol Vinyals, Quoc V. Le. “LSTM can solve hard long time lag problems.” 2014.

S. Hochreiter and J. Schmidhuber. “Sequence to Sequence Learning with Neural Networks.” 1997.

S. Hochreiter and J. Schmidhuber. “long short-term memory.” 1997.

European Parliament Proceedings Parallel Corpus 1996-2011 http://www.statmt.org/europarl/

Philipp Koehn “Europarl: A Parallel Corpus for Statistical Machine Translation”
http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdf.

https://coursefinders.com/es/blog/5669/cuantas-palabras-se-necesitan-para-hablar-con-fluidez-un-idioma

¿Cuántas palabras se necesitan para comunicarse?
https://www.bbc.com/mundo/noticias/2011/04/110330_palabras_ingles_lp

Robert Charles Lee “How many words are in the average English speaker's working vocabulary?”
https://www.quora.com/how-many-words-are-in-the-average-english-speakers-working-vocabulary

Tablas Retributivas UC3M
https://www.uc3m.es/ss/Satellite/RHPas/es/Detalle/Ficha_C/1371246114151/1371245242201/Tablas_Retributivas

REFERENCIAS

[1][3] Denny Britz, Anna Goldie, Minh-Thang Luong, Quoc Le, "Tutorial: Neural Machine Translation", [En línea]. Disponible en: https://google.github.io/seq2seq/nmt/.

[2][10] "European Parliament Proceedings Parallel Corpus 1996-2011" [En línea] Disponible en: http://www.statmt.org/europarl/

[4] "Reglamento General de Protección de Datos" [En línea] Disponible en: https://rgpd.es/

[5] Alison Kroulek "Google Translate Mistakes: 6 Times Google Went Rogue" 3/2/2016 [En línea] Disponible en: https://www.k-international.com/blog/google-translate-mistakes/

[6] S. Hochreiter and J. Schmidhuber. "Sequence to Sequence Learning with Neural Networks." 1997.

[7] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. "LSTM can solve hard long time lag problems." 2014.

[8] "Word2Vec, Doc2vec & GloVe: Neural Word Embeddings for Natural Language Processing" [En línea] Disponible en: https://deeplearning4j.org/docs/latest/deeplearning4j-nlp-word2vec

[9] S. Hochreiter and J. Schmidhuber. "long short-term memory." 1997.

[11] "Europarl: A Parallel Corpus for Statistical Machine Translation" [En línea] Disponible en: http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdf.

[12] "¿Cuántas palabras se necesitan para hablar con fluidez un idioma?" [En línea] Disponible en: https://coursefinders.com/es/blog/5669/cuantas-palabras-se-necesitan-para-hablar-con-fluidez-un-idioma

[13] "¿Cuántas palabras se necesitan para comunicarse?" BBC[En línea] Disponible en: https://www.bbc.com/mundo/noticias/2011/04/110330_palabras_ingles_lp

[14] "How many words are in the average English speaker's working vocabulary?" Robert Charles Lee [En línea] disponible en: https://www.quora.com/how-many-words-are-in-the-average-english-speakers-working-vocabulary

[15] "tf.nn.sparse_softmax_cross_entropy_with_logits" TensorFlow [En línea] disponible en: https://www.tensorflow.org/api_docs/python/tf/nn/sparse_softmax_cross_entropy_with_logits

[16] [En línea] dosponible en: https://www.uc3m.es/ss/Satellite/RHPas/es/Detalle/Ficha_C/1371246114151/1371245242201/Tablas_Retributivas

[17] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio "Neural Machine Translation by Jointly Learning to Align and Translate"

DETALLE DE FIGURAS