DISEÑO E IMPLEMENTACIÓN DE REDES NEURONALES RECURRENTES PARA LA TRADUCCIÓN AUTOMÁTICA

ANDRÉS ALBERTO PÉREZ BECERRA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

DISEÑO E IMPLEMENTACIÓN DE REDES NEURONALES RECURRENTES PARA LA TRADUCCIÓN AUTOMÁTICA

ANDRÉS ALBERTO PÉREZ BECERRA

Trabajo de grado para optar al título de
Ingeniero de Sistemas

DIRECTOR
Ph.D RAUL RAMOS POLLÁN
Escuela de Ingeniería de Sistemas e Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
BUCARAMANGA
2017

CONTENIDO

TABLA DE FIGURAS

LISTA DE TABLAS

Pag

RESUMEN

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE REDES NEURONALES RECURRENTES PARA LA TRADUCCIÓN AUTOMÁTICA.*

AUTOR: ANDRÉS ALBERTO PÉREZ BECERRA**

PALABRAS CLAVES: Redes neuronales recurrentes (RNN), Traducción automática neuronal (NMT)

DESCRIPCIÓN:

La traducción automática neuronal es un enfoque reciente en la traducción automática. El traductor pertenece a la familia del codificador-decodificador encontrado en varios artículos. Este campo aún tiene espacio para mejoras pero ha sido estudiado solo en los siguientes idiomas: inglés, francés y alemán.

En este proyecto se aplican las redes neuronales recurrentes para la traducción automática del español al inglés usando dos modelos, uno con 1000 unidades escondidas GRU (arquitectura 1) en cada RNN y otra usando 500 unidades escondidas GRU (arquitectura 2). Igualmente se usan dos *dataset* de entrenamiento, uno con 54M de palabras y otro con 28M de palabras y un *dataset* de prueba que consta de alrededor de 5M de palabras.

Se obtuvieron resultados favorables usando el *dataset* de 54M con la arquitectura 1 obteniendo un puntaje BLEU de 27.60 comparado al estado del arte de 28.45, la diferencia es que el estado del arte usa un *dataset* 6 veces mayor al de este proyecto. Igualmente la arquitectura 2 con el *dataset* de 54M de palabras demostró ser efectiva al obtener 24.74 puntos sin haber tenido el mismo número de iteraciones o tiempo de computo de las otros modelos.

Ha de resaltarse la capacidad de traducir por los modelos se basa en el uso de textos afines con los usados en el entrenamiento, ya que mantiene un orden sintáctico incluso cuando existen palabras desconocidas en la frase.

*Trabajo de grado

**Facultad de Ingenirias Fisico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Ph.D Raúl Ramos Pollan

ABSTRACT

TITLE: DESIGN AND IMPLEMENTATION OF RECURRENT NERUAL NETWORKS FOR MACHINE TRANSLATION.*

AUTHOR: ANDRÉS ALBERTO PÉREZ BECERRA. **

KEY WORDS: Recurrent neuronal networks (RNN), neural machine translation (NMT)

DESCRIPTION:

The neural machine translation is a recent approach in machine translation. The translator belongs to the coder-decoder family found in many articles. This field still show room for advances, but only has been studied in the following languages: English, French and German.

In this project the recurrent neural network are used for the machine translation form Spanish to English using two models, one with 1000 GRU hidden units (architecture 1) and the other with 500 GRU hidden units (architecture 2). Here two training dataset are used, one with 54M of words and the other with 28M of words, and a test dataset of 5M of words.

We got good results using the 54M dataset with the architecture 1 getting a BLEU score of 27.60, very similar to the state of the art score 28.45, the difference with the state of the art is that they use a dataset 6 times bigger than the used in this project. The architecture 2 with the 54M dataset showed a good performance getting 24.74 BLEU points without the same iteration number or time computed as of the other models.

It has to be said that the capacity to translate by the models is based in the use of related texts to those used in the training, in this way the translated sentence will keep a syntactic order even when there are words than it doesn't recognize in the sentence.

*Bachelor Thesis

**Faculty of Physical-Mechanical Engineering. School of Engineering and Computer Science. Director: Ph.D Raúl Ramos Pollan

INTRODUCCIÓN

Una parte de la vida humana se ha desarrollado en torno al idioma, un sistema de comunicación que varía entre distintas comunidades lo cual hace difícil el intercambio de ideas verbales, a excepción de ciertos individuos que manejan más de un idioma o lo suficiente de otro para hacer entender sus ideas. Esto mismo sucede con la escritura, es difícil de entender por gente ajena al lenguaje en el que está escrito el texto, incluso conseguir diferentes textos era difícil hasta hace unos años, donde muchos libros no tenían traducción.

En los últimos años la creación de editoriales en diferentes regiones e idiomas del mundo ha permitido el ingreso de diferentes libros en su idioma original tanto como con sus traducciones, estas traducciones son un proceso largo ya que en muchos casos dependen de personas expertas y múltiples revisiones antes de salir a la luz. Éste es el motivo de diferentes tecnologías que apoyan estos procesos.

La traducción automática nace como toda tecnología con la idea de resolver situaciones con la asistencia mínima de un operador humano, esta es la razón de diferentes enfoques en esta área como los métodos basados en reglas, donde se usan las reglas sintácticas y semánticas de los lenguajes objetivo y fuente al igual que tablas de palabras, esto consume muchos recursos y de igual forma, necesita un experto para la revisión de las reglas. Otro enfoque es la traducción por modelos estadísticos cuyos parámetros son derivados del análisis de textos bilingües.

Los modelos estadísticos son los más estudiados, así han sido introducidas las redes neuronales y el aprendizaje profundo para apoyar esta tarea. De esto nace la traducción automática neuronal. La traducción automática neuronal usa textos bilingües a la vez que redes neuronales para desarrollar los parámetros estadísticos, hasta ahora los artículos que han surgido sobre este tema intentan mejorar la calidad de la traducción pero han estado enfocados en los idiomas inglés, francés y alemán.

En este trabajo se busca usar la traducción automática neuronal con los idiomas español e inglés como idiomas fuente y objetivo respectivamente, observando sus resultados en diferentes textos.

1. DEFINICIÓN DEL PROBLEMA

En estos tiempos el flujo de información ha aumentado considerablemente, cada día existen nuevos avances en la ciencia y tecnología y la manera de acceder a esta información es facilitada por distintos medios como lo son revistas, libros y materiales electrónicos. El problema de muchas personas no ha sido el acceso a la información, sino una manera rápida y sencilla de entenderla debido a la barrera idiomática y aunque el número de personas que manejan una segunda lengua e institutos que apoyan este aprendizaje han aumentado, muchas personas aún no pueden entender textos en inglés o dar sus ideas en ingles de manera coherente.

Métodos tradicionales para traducir textos son de ayuda pero no son perfectos, muchas veces tienden a confundir a los usuarios, debido a que al ingresar frases muy largas estos métodos tradicionales no tienen en cuenta ambigüedades y traducen conforme a una tabla, dando a lugar traducciones que no son erradas pero tampoco son comprensibles.

Nuevos avances se están dando en este campo como lo es el paradigma de la traducción automática estadística, y más concretamente el uso de redes neuronales en este paradigma llamado traducción automática neuronal. La traducción automática neuronal ha avanzado pero hasta el momento se ha usado en los idiomas inglés, alemán y francés, no existe un trabajo con el español donde se demuestre su eficacia.

2. OBJETIVOS

2.1. **Objetivo general**

Diseñar, implementar y evaluar distintas arquitecturas de redes neuronales recurrentes para la traducción automática del español.

2.2. **Objetivos específicos**

- Explorar y definir *datasets* de prueba, los cuales consisten en colecciones de textos en un lenguaje L_1 y su traducción a un lenguaje L_2 (*parallel corpus*).
- Evaluar y seleccionar un *framework* para la implementación de redes recurrentes.
- Diseñar distintas arquitecturas de redes recurrentes para la traducción automática.
- Implementar las redes recurrentes sobre el *framework* seleccionado.
- Comparar el desempeño de las redes diseñadas con el estado del arte.

3. MARCO TEÓRICO

3.1. Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial que se define como un conjunto de métodos que pueden detectar automáticamente patrones en datos y usar dichos patrones para predecir datos futuros o para realizar otro tipo de decisión bajo incertidumbre [11]. Estos métodos operan construyendo modelos a partir de datos de entrada (datos de entrenamiento), los cuales no necesitan instrucciones estáticas para realizar una acción, esto se usa en muchas aplicaciones como: reconocimiento de caracteres, traducción automática, reconocimiento de rostros entre otras.

El aprendizaje automático está dividido generalmente en dos grandes tipos. El enfoque predictivo o de aprendizaje supervisado y el enfoque descriptivo o aprendizaje no supervisado.

3.1.1 Aprendizaje supervisado

Su meta es aprender a mapear las entradas x a las y dado un conjunto etiquetado de pares de entrada y salida tal que $D = \{(x_i, y_i)\}_{i=1}^{N}$. Donde D es el conjunto de entrenamiento y N es el número de ejemplos de entrenamiento.

De una manera simple cada entrada x_i es un vector de numeros de dimensión D que pueden representar distintos atributos o ser objetos complejos como imágenes, mensajes de correo electrónico, grafos, etc. Similarmente la forma de la salida puede ser en principio cualquier cosa, pero la mayoría de los métodos asumen que y_i es una variable categórica o nominal de un conjunto finito, donde y_i ϵ {1,…., C}, o que y_i es un valor real. Cuando la salida es categorica, el problema es conocido como clasificación o reconocimiento de patrones, y cuando y_i es un valor real, el problema es conocido como regresión.

3.1.1.1 Clasificación

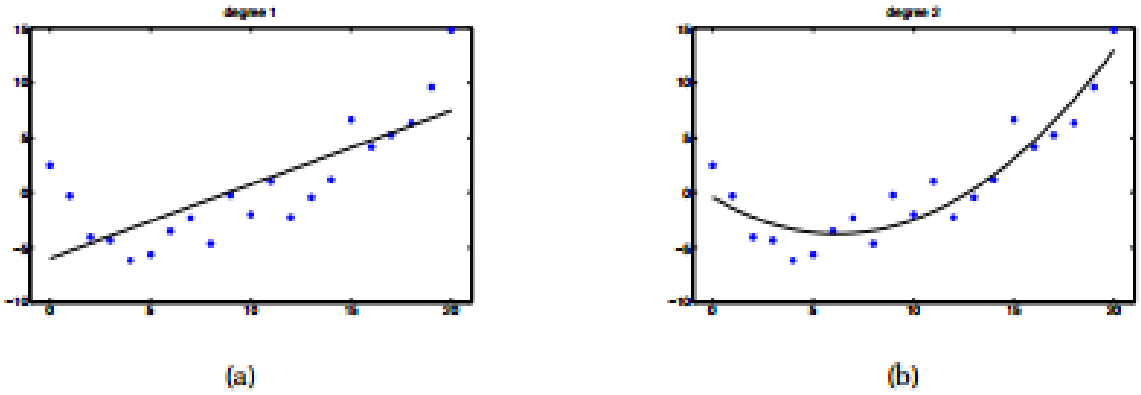
La meta de la clasificación es aprender a mapear una entrada x a una salida y, donde y ϵ {1,…., C}, con C como el número de clases. Así si C=2 la clasificación se llamara binaria y si C>2 se llamara clasificación multiclase. Si las etiquetas de las clases no son mutuamente exclusivas se llamara clasificación de múltiples etiquetas, pero esto es mejor visto como predicción relacionada a múltiples etiquetas de clase binaria.

Una manera de formalizar el problema es como una aproximación de funciones. Asumimos *y= f(x)* para una función desconocida *f*, y el objetivo es aprender a estimar la función *f* dado un conjunto de entrenamiento etiquetado y entonces hacer predicciones usando *y' = f'(x).* El objetivo principal es realizar predicciones con entradas que no se han visto antes.

3.1.1.2 **Regresión**

Regresión es justo como la clasificación, solo que la respuesta a la variable es continua. En la figura 1 se muestra un ejemplo simple donde existe una entrada $x_i \in R$ y una respuesta de valor real $y_i \in R$. Consideramos ajustar dos modelos a los datos: a) una línea recta y b) una función cuadrática.

Figura 1. Ejemplo de regresión



Fuente: Machine Learning A Probabilistic Perspective

3.1.2 **Aprendizaje no supervisado**

El segundo tipo es el enfoque descriptivo o aprendizaje no supervisado. En este enfoque solo tenemos como entradas $D = \{x_i\}_{i=1}^{N}$ y el objetivo es encontrar "patrones interesantes" en los datos. Este es un problema mucho menos definido ya que no se da el patrón a buscar y por tanto no hay una métrica de error para usar, a diferencia del aprendizaje supervisado donde se pueden comparar las predicciones de y para un valor dado x en un valor observado.

Para abarcar esta tarea se trabajara como una estimación de densidad, que es construir modelos de la forma $p(x_i|\theta)$. Hay dos diferencias con el caso supervisado. Primero, está escrito de la forma $p(x_i|\theta)$ en vez de $p(y_i|x_i|\theta)$; esto es debido a que el aprendizaje supervisado es una estimación de densidad condicional y el aprendizaje no supervisado es una estimación de densidad no condicional. Segundo, x_i es un vector de características, así que se necesitan crear modelos de probabilidad multivariables en contraste al aprendizaje supervisado donde y_i es usualmente una variable que se intenta predecir. Esto quiere decir que para la mayoría de casos de aprendizaje supervisado se pueden usar modelos de probabilidad de una variante que simplifican de gran manera el problema.

Aprendizaje no supervisado es posiblemente la manera más típica de aprender en humanos y animales. También es ampliamente más aplicable que el aprendizaje supervisado, ya que este no requiere de un experto para etiquetar los datos.

3.2 **Redes Neuronales Artificiales**

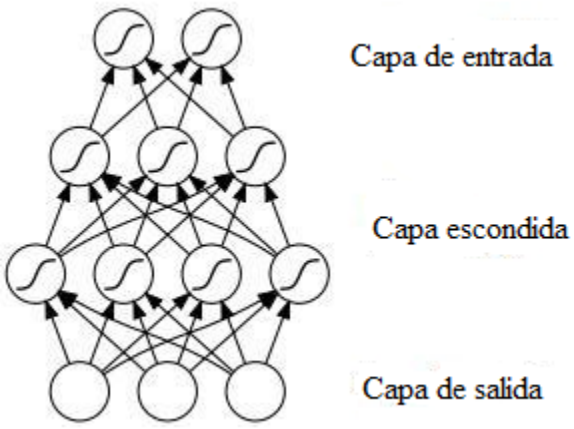
Las redes neuronales artificiales (ANN por sus siglas en inglés) fueron desarrolladas originalmente como modelos matemáticos de la capacidad de procesar información en cerebros biológicos. Aunque ahora es claro que las ANN poseen poca semejanza con las neuronas biológicas reales, estas poseen una amplia popularidad como clasificadores de patrones.

La estructura básica de una ANN es una red de pequeñas unidades de procesamiento o nodos, unidas entre ellas por conexiones ponderadas. En términos del modelo biológico original los nodos representan neuronas y las conexiones ponderadas representan la fuerza de las sinapsis entre las neuronas. La red es activada al proveer una entrada a algunos o todos los nodos, y su activación es entonces propagada a través de la red a lo largo de las conexiones ponderadas.

Gran variedad de ANNs han surgido a lo largo de los años con diferentes propiedades. Una distinción importante entre las ANN es el tipo de conexión, están las que sus conexiones son cíclicas y las que son acíclicas, las primeras son conocidas como de retroalimentación, recursivas o recurrentes y las acíclicas son llamadas redes de propagación hacia adelante (FNN por sus siglas en ingles).

Las unidades en una red FNN están organizadas en capas con conexiones hacia delante de una capa a la siguiente. Patrones son presentados en la capa de entrada, propagados a través de la capa escondida (esta capa se encarga del procesamiento interno en la red) hacia la capa de salida como se ve en la figura 2. Este proceso es conocido como *paso hacia delante* de la red.

Figura 2. Organización de una ANN



3.2.1 **Redes neuronales recurrentes**

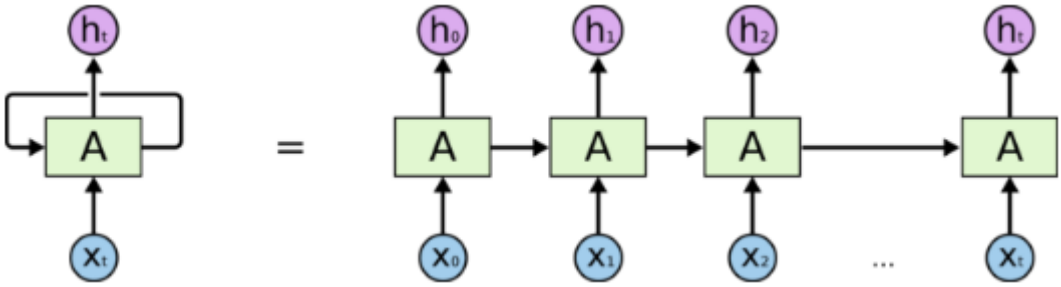
Las redes neuronales recurrentes (RNN por sus siglas en inglés) son aquellas cuyas conexiones son cíclicas, y aunque parece una diferencia sencilla con las FNN sus implicaciones de aprendizaje son de mayor alcance.

Las RNN son poderosos modelos de aprendizaje que logran resultados de estado de arte en una amplia gama de tareas de aprendizaje supervisado y no supervisado. Son especialmente diseñadas para tareas de percepción, donde las características puras subyacentes no son interpretadas individualmente [8]. Algunos ejemplos de su uso se pueden encontrar en las tareas de reconocimiento de discurso, modelamiento de lenguaje, traducción y subtitulado de imágenes.

Estos modelos de aprendizaje están caracterizados por su capacidad de en principio mapear la historia de las entradas anteriores a cada salida, en otras palabras las conexiones recurrentes en una red permiten una "memoria" de entradas anteriores que persisten en el estado interno de la misma y por tanto influencian la salida de la red.

Considerando una parte de una red recurrente A cuya entrada es x_t y una salida h_t, un bucle permite que la información siga de un paso de la red al siguiente. Otra manera de ver esto es considerar la red como múltiples copias de la misma, cada una pasando un mensaje a su sucesor como se ve en la figura 3.

Figura 3. Red recurrente



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

La información pasa por la capa escondida antes de regresar en el bucle como una entrada. El estado escondido h_t en un paso de tiempo es función de la entrada en el mismo paso de tiempo x_t, modificada por una matriz de peso W, más el producto del estado escondido en el paso de tiempo anterior h_{t-1} por su matriz de estado escondido a estado escondido U conocida como matriz de transición. El error que ellos generan regresa usando *propagación hacia atrás*, lo cual ajusta los pesos hasta que el error no puede bajar más. La siguiente ecuación muestra el estado escondido en un tiempo t.

$$h_t = \theta(Wx_t + Uh_{t-1})$$

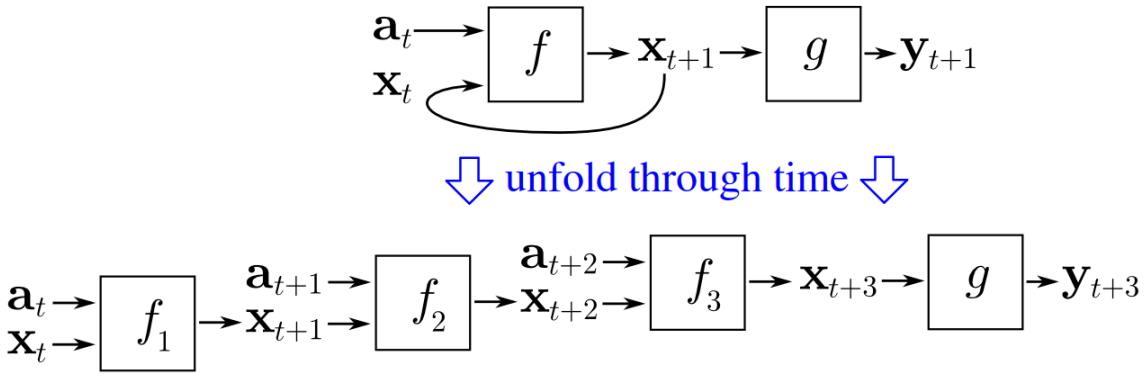
La suma de los pesos de entrada y el estado escondido está condesando por la función θ la cual es una herramienta para condensar valores muy grandes o muy pequeños en un espacio logístico, a la vez que hacen las gradientes adecuadas para la propagación hacia atrás.

3.2.1.1 **Propagación hacia atrás a través del tiempo**

La propagación hacia atrás en redes que se alimentan hacia adelante, se basa en que una vez que se ha obtenido una salida esta se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Las salidas se propagan hacia atrás partiendo por la capa de salida, llegan a la capa escondida donde los pesos de las neuronas son actualizados dependiendo de la señal del error y de la contribución relativa a la salida.

Redes recurrentes se basan en una extensión de la propagación hacia atrás llamada propagación hacia atrás a través del tiempo (BPTT por sus siglas en ingles). En este caso una serie de cálculos conectan un paso de tiempo con el siguiente, esto empieza desplegando una red recurrente a través del tiempo como se muestra en la figura 4. Dicha red recurrente contiene dos redes que avanzan hacia adelante, *f* y *g*. Cuando la red se despliega a través del tiempo, esta contiene *k* instancias en *f* y una en *g*.

Figura 4. Propagación hacia atrás.



Fuente: https://en.wikipedia.org/wiki/File:Unfold_through_time.png

Después procede un entrenamiento en el que por cada *epoch* se miran todas las observaciones y_t en orden secuencial. Cada patrón de entrenamiento consiste de $(x_t,a_t,a_{t+1},a_{t+2},\ldots,a_{t+k-1},y_{t+k})$.

Desde de que cada patrón es presentando los pesos son actualizados, los pesos en cada instancia de *f* son $(f_1,f_2,\ldots,f_k)$, son promediados de tal forma que todos tengan el mismo peso. Igualmente x_{t+1} es calculado lo que provee información necesaria para que el algoritmo pueda seguir con el siguiente paso de tiempo $t+1$. x_{t+1} es calculado como:

$$x_{t+1}=f(x_t,a_t)$$

3.2.1.2 **Problema del gradiente**

La gradiente expresa el cambio en todos los pesos con respecto al cambio en el error, así, si no se conoce la gradiente no se pueden ajustar los pesos en una dirección en que el error decrezca.

Con el método convencional de propagación hacia atrás a través del tiempo la señal de error que fluye hacia atrás tiende a explotar o desvanecerse, ya que la evolución temporal del error de propagación depende exponencialmente del tamaño de los pesos [7]. En el caso donde los pesos explotan causa que los pesos oscilen; por su parte cuando los pesos tienden a desvanecerse el tiempo de entrenamiento aumenta considerablemente y en algunos casos el entrenamiento no funciona en lo absoluto. Existen soluciones cuando la propagación del error tiende a explotar, lo cual es truncar dicha propagación, pero no existía una solución al desvanecimiento es por esta razón que se crea una arquitectura de red recurrente llamada *long short-term memory* (LSMT) en el año de 1997.

3.2.1.3 **LSMT**

En 1997 la variación a las redes recurrentes llamadas LSMT fue propuesta por los investigadores Sepp Hochreiter y Juergen Schmidhuber como una arquitectura de red que en conjunción con un algoritmo apropiado basado en la gradiente puede sobreponerse al error de la gradiente descendiente[6].

La arquitectura en una LSTM permite que el error se propague hacia atrás a través de las capas y el tiempo, esto es gracias a una estructura nueva llamada *célula de memoria*.

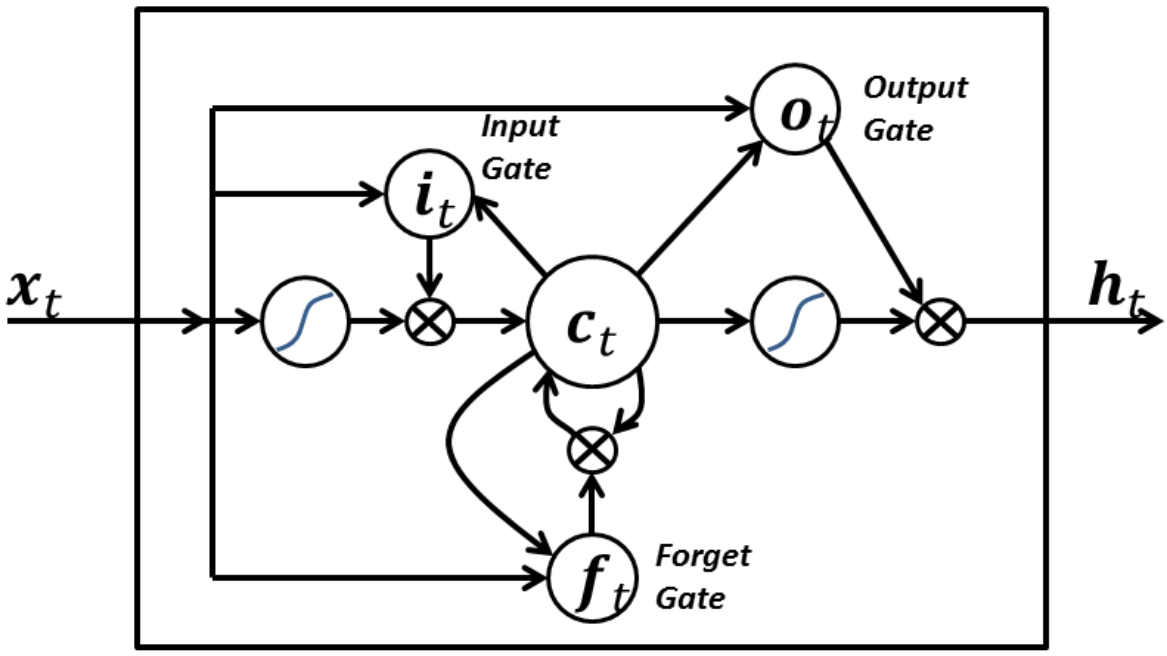
Una célula de memoria está compuesta por tres compuertas principales: una compuerta de entrada, una compuerta de olvido y una compuerta de salida, estas se pueden ver en la figura 5. Las compuertas sirven para modular la interacción entre la célula de memoria y su ambiente, la compuerta de entrada permite a señales entrantes alterar o no el estado de la célula de memoria. Por otra parte, la compuerta de salida permite al estado de memoria de la célula afectar a otras neuronas. Por último la compuerta de olvido puede modular la memoria de la célula, permitiendo que recuerde u olvide sus estados anteriores.

Células de memoria compartiendo la misma compuerta de entrada y de salida forman una estructura llamada bloques de células de memoria de tamaño S. Bloques de células de memoria facilitan el almacenamiento de la información. Como los bloques de células de memoria tienen tantas unidades de compuerta como células de memoria individuales, la arquitectura de bloque puede ser un poco más eficiente.

El algoritmo LSMT es bastante eficiente, con una complejidad de actualización de O(W), donde W es el número de pesos. Por tanto LSMT y BPTT para redes totalmente recurrentes tienen la misma complejidad de actualización por paso de tiempo. Pero a diferencia de BPTT, LSMT es local tanto en tiempo como en espacio, ya que su complejidad de actualización no depende del tamaño de la red (local en el espacio) y sus requerimientos de almacenamiento no dependen de la secuencia de entrada (local en el tiempo). LSMT no

necesita almacenar valores de activación observados durante pre procesamiento de secuencias en una pila.

Figura 5. **LSTM**



Fuente: https://en.wikipedia.org/wiki/File:Long_Short_Term_Memory.png

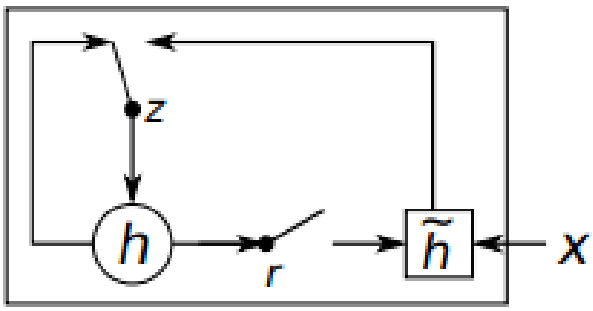
3.2.1.4 **GRU**

Esta unidad escondida llamada GRU por sus siglas en inglés (*gated recurrent unit*) es motivada por la unida LSTM pero es más simple de computar e implementar [4]. La figura 6 muestra la representación gráfica de la unidad, donde la compuerta de actualización *z* selecciona si el estado escondido es actualizado con un nuevo estado escondido $\tilde{h}$. La compuerta de reinicio *r* decide si los estados escondidos anteriores son ignorados.

A continuación se describirá como la compuerta de reinicio r_j es computada

$$r_j = \sigma([W_r x]_j + [U_r h_{(t-1)}]_j)$$

Figura 6. Ilustracion de la funcionde activación de la unidad escondida GRU



Fuente: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

Donde σ es la función lógica sigmoidal, y $[.]_j$ denota el j-esimo elemento del vector. x y h_{t-1} son la entrada y el estado escondido anterior respectivamente. W_r y U_r son matrices de pesos que son aprendidas. Similarmente la compuerta de actualización z_j es computada por

$$z_j = \sigma([W_z x]_j + [U_z h_{(t-1)}]_j)$$

La activación de la unidad propuesta h_j es computada de la siguiente manera

$$h_j^{\langle t \rangle} - z_j h_j^{\langle t-1 \rangle} + (1 - z_j)\tilde{h}_j^{\langle t \rangle}$$

Donde

$$\tilde{h}_j^{\langle t \rangle} - \phi\left([\mathbf{W}\mathbf{x}]_j + \left[\mathbf{U}\left(\mathbf{r} \odot \mathbf{h}_{\langle t-1 \rangle}\right)\right]_j\right)$$

En esta formulación cuando la compuerta de reinicio es cercana a 0, el estado escondido es forzado a ignorar el anterior estado escondido y reiniciar con la entrada actual solamente. Esto permite que el estado escondido libere efectivamente información que no es relevante en el futuro, permitiendo así una representación más compacta.

Por su parte la compuerta de actualización controla cuanta información del estado escondido anterior se llevará hasta el siguiente estado escondido. Como cada unidad escondida está separada de las compuertas de reinicio y actualización, cada unidad escondida aprenderá a capturar las dependencias en diferentes escalas de tiempo. Estas unidades que aprenden a capturar dependencias a corto plazo tienden a tener compuertas de reinicio que son frecuentemente activas, pero aquellas que capturen dependencias a largo plazo tendrán compuertas de actualización más activas.

3.3 **Traducción automática**

La traducción automática (MT por sus siglas en inglés) es un subcampo de la lingüística computacional que investiga el uso de software para la traducción de texto o discursos de un lenguaje a otro.

En un nivel básico MT realiza una sustitución simple de palabras de un lenguaje a otro, pero esto no puede producir una buena traducción de un texto porque el reconocimiento de la frase completa y su contraparte más cercana en el texto objetivo es necesitada. Resolviendo este problema han nacido técnicas estadísticas de rápido crecimiento.

Los actuales software de MT se enfocan generalmente en un dominio o campo lo cual aumenta la calidad de la salida al limitar la cantidad de posibles sustituciones. Esta técnica es particularmente efectiva en dominios donde lenguaje formal o donde son usadas formulas. Aun así existen diferentes métodos para la traducción automática, a continuación se presentaran algunos de ellos.

3.3.1 **Traducción automática basada en reglas**

Este tipo de traducción se basa mayormente en la creación de diccionarios y programas de gramática. Este método involucra abundante información sobre la lingüística del lenguaje fuente y el objetivo, usando las reglas morfológicas y sintácticas y análisis semántico de los dos lenguajes. El enfoque básico involucra la vinculación de la estructura de la frase de entrada con la estructura de la frase de salida usando un analizador para el lenguaje fuente, un generador para el lenguaje objetivo y un léxico de transferencia. Su más grande desventaja es la necesidad de hacer explícito todo, por ejemplo se deben escribir las reglas léxicas para todos los casos de ambigüedad.

3.3.2 **Traducción automática basada en ejemplos**

Propuesta por Makoto Nagao en 1984, este enfoque está basado en la analogía en el *parallel corpus*. Dada una frase que ha de ser traducida, se seleccionan frases del corpus que contengan componentes similares subescenciales. Las frases similares son usadas para traducir los componentes subescenciales de la frase original a la frase objetivo, y se juntan para completar la traducción.

3.3.3 **Traducción automática estadística**

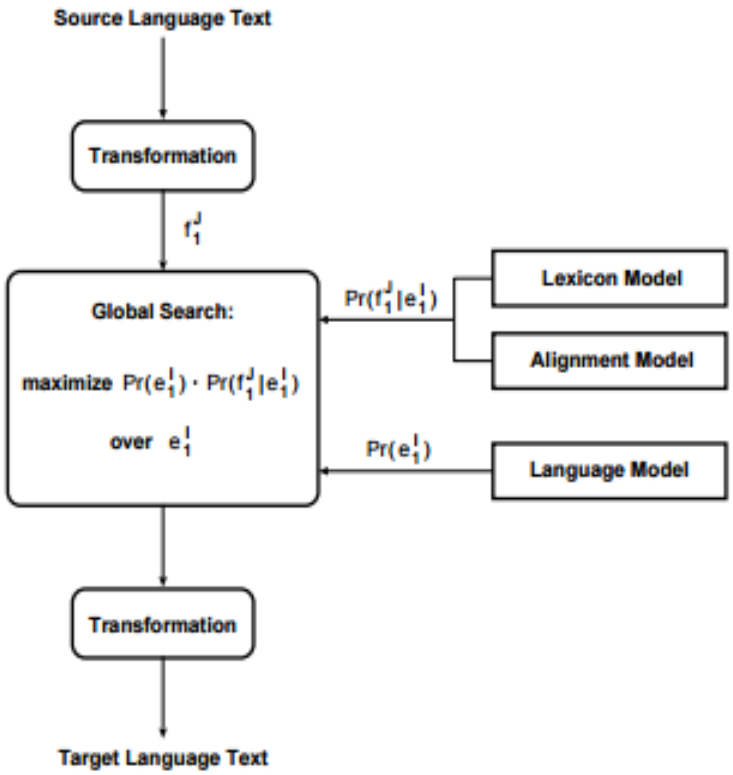
Un documento es traducido de acuerdo a la distribución de probabilidad p(e|f) dada una cadena fuente $f_1^J = f_1 \dots f_j \dots f_J$ la cual va a ser traducida en una cadena objetivo $e_1^J = e_1 \dots e_j \dots e_J$. Entre todas las posibles cadenas objetivo, se escogera la que tenga la mayor probabilidad de acuerdo a la regla de decisión de Bayes:

$$\hat{e}_1^I = argmax_{\hat{e}_1^I}\{Pr(e_1^I|f_1^J)\}$$

$$\hat{e}_1^I = argmax_{e_1^I}\{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\}$$

Aquí Pr(e_1^I) es el modelo de lenguaje del lenguaje objetivo y Pr($f_1^J|e_1^I$) es la cadena del modelo de traducción. El argumento argmax denota el problema de búsqueda. En la figura 7 se muestra en resumen la arquitectura general del enfoque de traducción estadística.

Figuta 7. Arquitectura general del SMT



Fuente: https://www-i6.informatik.rwth-aachen.de/web/Research/machine_trans.html

Para una rigurosa implementación de las formulas anterior se de realizar una búsqueda exhaustiva al ir por todas las cadenas e* en el lenguaje nativo. Realizar esta búsqueda eficientemente es el trabajo del decodificador que usa diferentes métodos para limitar el espacio de búsqueda y al mismo tiempo mantener una calidad aceptable.

3.3.3.1 **Traducción automática neuronal**

La traducción automática (NMT por sus siglas en inglés) neuronal es un enfoque de SMT basado únicamente en el uso de redes neuronales, estos usan generalmente un modelo de codificador-decodificador, donde el codificador extrae una representación vectorial de tamaño fijo de una frase de entrada de tamaño variable y a partir de esta representación el decodificador genera una traducción de tamaño variable.

Una NMT modela directamente la probabilidad condicional p(y|x) de traducir una frase, x1,…,xn, a una frase objetivo, y1,…,yn. Un codificador computa una representación s por cada frase fuente y un decodificar genera una palabra objetivo y por tanto descompone la probabilidad condicional como:

$$log\ p(\text{y}|\text{x}) = \sum_{j-1}^{m} \log p(\ (y_j|y_{<j}, s)$$

Una elección de modelo para la descomposición de este decodificador es usar una arquitectura de redes neuronales recurrentes. En más detalle se puede parametrizar la probabilidad de decodificar cada palabra y_j como:

$$p(y_j|y_{<j},s)=\text{softmax}(g(h_j))$$

Donde g es una función de transformación que produce un vector del tamaño del vocabulario. Acá, h_j es la unidad escondida de la RNN, computada abstractamente como:

$$h_j=f\ (h_{j-1},\ s)$$

Donde *f* computa el estado escondido actual dado el anterior estado escondido. La representación fuente *s* es solo usada una vez para inicializar el estado escondido del codificador.

3.3.4 **Puntaje BLEU para la traducción automática**

La evaluación de la traducción automática pesa muchos aspectos de la traducción, tales como la adecuación, fidelidad y fluidez de la traducción; estos aspectos pueden tardar hasta meses por un evaluador humano generando un problema al ser necesarios estos resultados para el mejoramiento de sistema.

La tarea principal de BLEU es comparar los n-gramos de la traducción candidata con los n-gramos de la traducción de referencia y contar el número coincidencias. Estas coincidencias son independientes de la posición [10].

3.3.4.1 **Precisión n-gramo modificado**

Para mejorar la precisión de la computación se podría simplemente contar el número de palabras candidatas de traducción (unigramos) que ocurren en cualquier traducción de referencia y entonces dividirlo por el número total de palabras en la traducción candidata. Desafortunadamente los sistemas MT generan palabras “razonables” pero improbables. Una manera de solucionar esto es agotar las palabras de referencia después de que una palabra candidata coincidente es identificada, esto será llamada unigramo de precisión modificado. Para computar esto se contaran primero el número máximo de veces que una palabra ocurre en cualquier traducción de referencia, seguido de recortar el conteo total de

cada palabra candidata por el conteo máximo de su referencia, sumar estos conteos recortados y dividirlo por el número total de palabras candidatas sin recortar.

En un *parrallel corpus* el método de evaluación es similar, ya que la unidad básica de evaluación es la frase. Primeramente se computa los n-gramos coincidentes frase a frase. Después se agrega la cuenta de los n-gramos cortados para todas las frases candidatas y las dividimos por el número de n-gramos candidatos en el corpus para computar un puntaje p_n para el corpus entero:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

3.3.4.2 **Penalidad en brevedad de la frase**

Se introduce un factor de penalidad para traducciones candidatas más largas que sus referencias, con esta penalidad traducciones candidatas deben coincidir con la traducción de referencia tanto en longitud, elección de palabras y orden de palabras. Esta penalidad es de 1.0 cuando la longitud de la traducción candidata es igual a su referencia.

Esta penalidad se computa sobre todo el corpus y no frase a frase, primeramente se obtiene la longitud efectiva de la referencia *r* al sumar las mejores coincidencias de longitud por cada grase candidata en el corpus. La penalidad de brevedad es un exponencial decreciente en *r/c*, donde *c* es el la longitud total de los candidatos de traducción del corpus.

3.3.4.3 **Evaluación del corpus**

Primeramente se computa el promedio geométrico de la precisión modificada n-gramos,pn, usando n-gramos hasta la longitud N y pesos positivos w_n sumando hasta uno. Después siendo *c* la longitud de la traducción candidata y *r* la longitud de la referencia efectiva del corpus. Se computa la penalidad de brevedad BP

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

Entonces:

$$\mathrm{BLEU} = \mathrm{BP} \bullet \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Las métricas BLEU van de 0 a 1 siendo 0 la peor traducción y 1 la mejor traducción igual a la referencia. Pocas traducciones acertaran un puntaje de 1 a menos que sean idénticas a la traducción de referencia, por esta razón un traductor humano no anotara necesariamente 1[10]. En este proyecto se trabajara esta métrica de 0 a 100 manteniendo el 0 como una traducción que no tiene nada ver con la referencia y 100 como una traducción igual a la referencia, la razón de esto es para comparar más fácil en la etapa de resultados con otros trabajos, ya que la mayoría de artículos trabajan esta métrica en este rango.

4. ESTADO DEL ARTE

La traducción automática estadística (SMT por sus siglas en inglés) es un paradigma de la traducción automática donde se generan traducciones en base a modelos estadísticos derivados del análisis de textos bilingües. Las primeras ideas fueron introducidas en 1949 por Warren Weaver y reintroducidas a finales de 1980 y a principios de 1990 por investigadores del centro de investigación Thomas J. Watson de la IBM quienes crearon el sistema CANDIDE, este sistema usaba métodos de teoría de información y estadística para desarrollar un modelo de probabilidad del proceso de traducción[2].

La traducción estadística es el paradigma de traducción más estudiado desde el 2006, el cual tiene nuevos avances como lo es la traducción automática neuronal, el cual nace entre 2013 por Kalchbrenner y Blunsom quienes proponen un modelo llamado modelo recurrente de traducción continua basado en representaciones para palabras, frases y oraciones, su idioma fuente es el inglés y el objetivo es francés[3]; 2014 por Sutskever quien presenta un enfoque de aprendizaje secuencia a secuencia que realiza asunciones mínimas en la estructura de la secuencia, usa una LSTM para mapear la secuencia de entrada a un vector de dimensionalidad fija y otra LSTM para decodificar la secuencia del vector, su idioma fuente es inglés y el objetivo es francés[12]; y en el mismo año Cho propone su modelo llamado codificador-decodificador RNN, donde una RNN codifica una secuencia de símbolos en un vector de tamaño fijo y otra RNN decodifica la representación en otra secuencia, igualmente sus lenguajes corresponden al inglés y francés como idiomas fuente y objetivo respectivamente[4].

El modelo codificador-decodificador es usado para este proyecto, donde diferentes pruebas se han realizado con este modelo como son el uso de redes convolucionales en vez de redes recurrentes y la comparación de su desempeño con los idiomas inglés y francés [5]. Otra prueba ha sido un mecanismo de atención implementado al modelo, el cual se concentra en ciertas partes de la frase fuente, dos clases de mecanismo fueron desarrollados: un enfoque global que trabaja con todas las palabras fuente y una local que solo mira un subconjunto de palabras fuente a la vez; esta tarea fue aplicada a la traducción inglés-alemán y alemán-ingles [9].

El modelo de Cho anteriormente mencionado tiene la debilidad de ser incapaz de traducir correctamente palabras raras, ya que los sistemas NMT tienen vocabularios relativamente pequeños con un solo símbolo *unk* que representa las palabras fuera del diccionario, por esto se entrena la NMT con un algoritmo de alineamiento que por cada palabra *unk* en la frase objetivo se referencia su posición en la frase fuente, esta información es usada para un proceso de post-procesamiento que traduce cada palabra desconocida usando un diccionario, este trabajo utilizo como idiomas fuente y objetivo el inglés y el francés respectivamente[1].

5. METODOLOGIA

El proyecto se basó en la metodología ágil KANBAN, la cual se resume en que el trabajo debería limitarse a lo que existe y no se empieza otro bloque u otra función de la cadena hasta no haber terminado el actual.

Con esta metodología se busca que cada paso dado fuera estudiado de la mejor manera para no tener que volver en un futuro a realizar correcciones sobre ello. Los pasos realizados fueron los siguientes:

1. Estudio del estado del arte: primeramente se buscaron artículos y blogs sobre NMT y RNN ya que son el eje central de este proyecto, a medida que el estudio avanzaba se iba complementando los temas como la traducción automática y la manera de medir los resultados de las traducciones.

2. Selección de herramientas: este punto es tratado en el desarrollo del proyecto y en el cual nos enfocamos un tiempo porque aunque el cambio de un *framework* a otro no es muy diferente si consume tiempo, por esta razón sobre todo la selección de *framework* fue estudiado con el ánimo de crear el traductor.

3. Redes neuronales recurrentes: al tener el lenguaje y *framework*, se buscaron ejemplos de redes neuronales para entender su funcionamiento en el código y codificación de palabras.

4. NMT: se decide por el modelo de red a trabajar y su posterior codificación.

5. *Dataset*: este paso es conjunto al diseño de la red y es el único caso en el que hubo estudio hasta el final, ya que *parallel corpus* en inglés-español son escasos y deben de tener cierto tamaño para que la red se entrene satisfactoriamente.

6. Pruebas y resultados: se pasa a iterar los *dataset*, lo cual lleva una cantidad de tiempo explicada en el desarrollo del proyecto y posteriormente analizar los resultados obtenidos.

7. Documentación: Al obtener resultados acertados se procede a documentar los puntajes y tiempos.

6. DESARROLLO DEL PROYECTO

En este capítulo se presentara el proceso del diseño de la red empleada para la traducción español-inglés usando redes neuronales recurrentes, el proyecto se desarrolló en 4 fases principales:

- Selección de herramientas
- Pre procesamiento del corpus
- Diseño y entrenamiento de modelo
- Evaluación del modelo

6.1 **Selección de herramientas**

A continuación describiremos el proceso que seguimos para decidir que herramientas usar en la tarea de desarrollar el traductor automático neuronal.

6.1.1 **Selección de lenguaje**

La construcción de las redes neuronales recurrentes se puede realizar en diferentes lenguajes de programación como c++, matlab y python entre otros, así mismo diferentes *frameworks* son desarrollados para dichos lenguajes.

Primeramente el lenguaje de programación elegido fue python, esto nos ofrece un lenguaje soportado en diferentes sistemas y plataformas, además de ser fácil de aprender, leer y manejar. Igualmente python es desarrollado bajo la licencia de "fuente abierta" lo cual lo hace libre de usar y distribuir incluso para uso comercial, esta misma es la razón de sus diferentes *frameworks* desarrollados y retroalimentados por la comunidad.

6.1.2 **Selección Framework**

Una vez escogido el lenguaje de programación un *framework* fue seleccionado para el diseño de las RNN enfocadas a la traducción automática, los siguientes son algunos de los *frameworks* estudiados.

6.1.2.1 **Framework Keras**

Keras[1] es una librería de redes neuronales escrita en python y puede funcionar tanto en Theano como TensorFlow. Keras está diseñado para las siguientes necesidades:

- Prototipado fácil y rápido
- Diseño de redes convolucionales y recurrentes, al igual que la combinación de ambas
- Puede correr tanto en CPU como en GPU

[1] https://keras.io/

Este *framework* es fácil de usar pero no permite construir y entrenar modelos de redes neuronales secuencia a secuencia, son estos modelos los usados para la traducción automática y por tanto Keras aunque fue estudiado no fue aplicado para el desarrollo de este proyecto.

6.1.2.2 **Framework Groundhog**

Groundhog[2] es un *framewoork* desarrollado en la universidad de Montreal y entre sus muchos colaboradores están Razvan Pascanu, KyungHyun Cho y Caglar Gulcehre. Este *framework* es construido sobre Theano y puede ser usado con Jobman y provee una manera flexible y eficiente de implementar modelos recurrentes complejos. Entre esos modelos complejos podemos encontrar el codificador-decodificador RNN.

Este fue el *framework* escogido por su capacidad de generar las RNN que necesitamos además de que sus creadores como KyungHyun[3] han presentado artículos sobre traducción automática muy seguramente usando este *framework*.

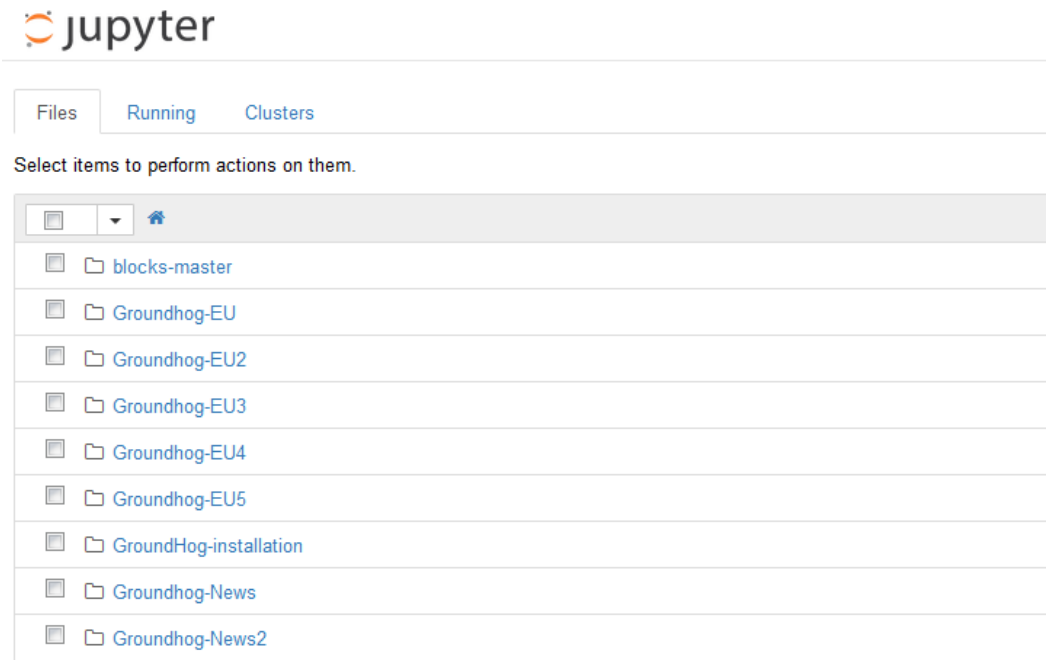
6.1.3 **Organización de la herramienta**

Ya escogido el lenguaje y el *framework* para el diseño e implementación de las redes seguiremos con la aplicación donde se llevara a cabo. Para esta tarea Jupyter[4] es una herramienta web adecuada, ya que soporta diferentes lenguajes de programación al igual que la creación y participación de documentos que contengan código, ecuaciones, visualizaciones y texto. Jupyter me permite la navegación en carpetas (como se ve en la figura 8) y "notebooks" desde los cuales hacemos llamadas a las funciones, esto permite compartir información y tareas de una manera sencilla y práctica.

En el caso del este proyecto Jupyter esta soportado por GUANE, un cluster de la UIS compuesto de 16 nodos ProLiant SL390s G7.

[2] https://github.com/pascanur/GroundHog
[3] www.kyunghyuncho.me
[4] http://jupyter.org/

Figura 8. Diferentes carpetas usadas en Jupyter.

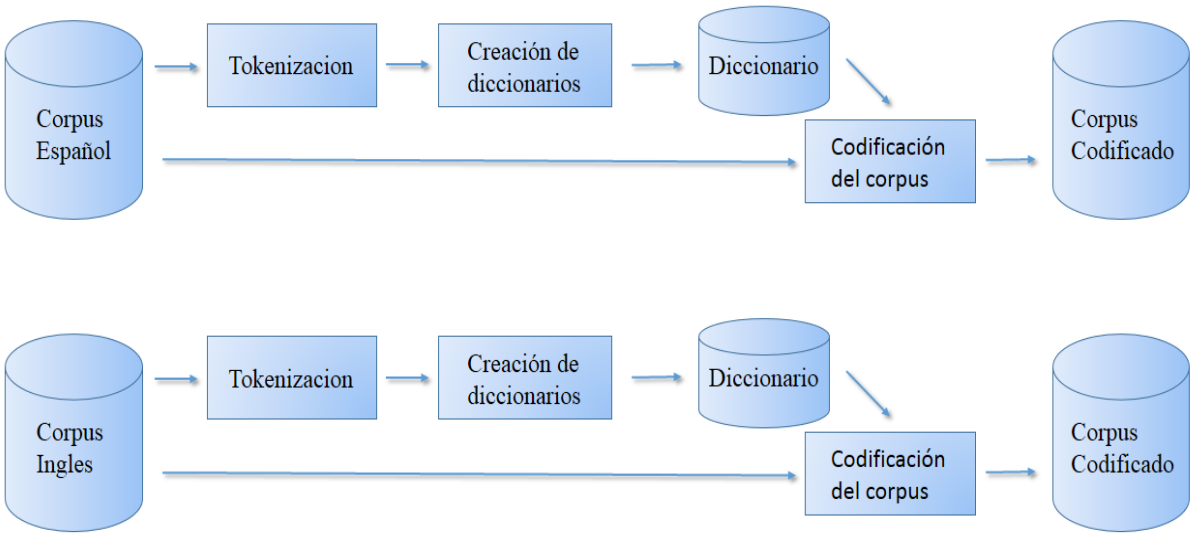


6.2 **Pre procesamiento de los corpus**

Para el desarrollo de proyecto se necesita de un texto en el lenguaje fuente y su traducción en el lenguaje objetivo, aun así estos textos deben ser lo suficientemente grandes para realizar el entrenamiento de la red, estos *parallel corpus* son difíciles de encontrar para el idioma de inglés-español, aun así los textos provistos por ACL WMT '14[5] contienen textos con los requerimientos buscados. Los siguientes *parallel corpora* de español-inglés son los encontrados: Europarl (54M palabras), news commentary (5,5M palabras) y UN (327M palabras), los escogidos son Europarl y news commentary, al igual que una fracción de Europarl con 28 M de palabras. A continuación como se ve en la figura 9, se presentan los pasos realizados para el pre procesamiento.

[5] http://www.statmt.org/wmt14/translation-task.html

Figura 9. Pre-procesamiento



6.2.1 **Tokenizacion**

Dada una secuencia de caracteres de entrada, la tokenizacion es la tarea de cortar dicha secuencia en piezas llamadas *tokens*, a la vez que puede desechar ciertos caracteres.

Generalmente la tokenizacion ocurre a nivel de palabras, aunque es difícil definir lo que una palabra significa, por esta razón al realizar el proceso de tokenizacion se apoyan en ciertos heurísticos, por ejemplo:

- Puntuaciones y espacios en blanco pueden o no ser incluidos en el resultado final.
- Todas las cadenas continuas de caracteres alfabéticos son parte de un solo token, al igual que con los números.
- Los tokens son separados por caracteres de espacio en blanco, tales como espacio o saltos de línea, o por caracteres de puntuación.

Como se explicó anteriormente este proceso separa la secuencia de caracteres de entrada de los documentos ingresados en unidades de procesamiento llamados *tokens*. Aunque al revisar los documentos no parezca haber un cambio demasiado grande, este es un proceso necesario y el cual requiere de documentos de los prefijos no separables en ambos idiomas. Tanto los documentos con los prefijos no separables y el script de tokenización se obtuvieron de MOSES[6].

[6] http://www.statmt.org/moses/

Figura 10. Tokenizacion

```
In [1]: !perl preprocess/tokenizer.perl -l en < data/Ingles.txt > data/Ingles.tok.txt
        !perl preprocess/tokenizer.perl -l es < data/Español.txt > data/Espanol.tok.txt

        Tokenizer Version 1.1
        Language: en
        Number of threads: 1
        Tokenizer Version 1.1
        Language: es
        Number of threads: 1
```

Ha de notarse que se dijo que el código escogido fue python, pero estos scripts ya están diseñados en Perl y demuestra la flexibilidad de Jupyter como herramienta de trabajo como se muestra en la figura 10.

6.2.2 **Creación de diccionarios**

Para el entrenamiento se usan las 30000 palabras más frecuentes tanto para inglés como español, así creando un diccionario asignando a cada palabra un id, las palabras que no se encuentren en este diccionario serán mapeadas a un *token* especial llamado UNK. Se mostrara los valores correspondientes de los diccionarios en cada *parallel corpora*:

- Europarl: El texto de inglés contiene 132904 palabras únicas en 1965734 frases con un total de 54506401 de palabras, el diccionario con las 30000 palabras cubre un 99.5% del texto. El texto en español contiene 195307 palabras únicas en 1965734 frases con un total de 57050736 palabras, el diccionario cubre un 98.8% del texto.

- Europarl fracción: El texto de inglés contiene 95088 palabras únicas en 1000000 frases con un total de 27858619 de palabras, el diccionario con las 30000 palabras cubre un 99.5% del texto. El texto en español contiene 140820 palabras únicas en 1000000 frases con un total de 28923828 palabras, el diccionario cubre un 98.8% del texto.

Al haber creado el diccionario crearemos su inverso, un diccionario donde a cada id se le asigna una de las 30000 palabra más frecuentes como se muestra en la figura 11.

Figura 11. Creación de diccionarios

```
In [2]: !python preprocess/preprocess.py -d data/vocab.en.pkl -v 30000 -b data/binarized_text.en.pkl -p data/Ingles.tok.txt
        !python preprocess/preprocess.py -d data/vocab.es.pkl -v 30000 -b data/binarized_text.es.pkl -p data/Espanol.tok.txt
        !python preprocess/invert-dict.py data/vocab.en.pkl data/ivocab.en.pkl
        !python preprocess/invert-dict.py data/vocab.es.pkl data/ivocab.es.pkl

        INFO:preprocess:Counting words in Ingles.tok.txt
        INFO:preprocess:95088 unique words in 1000000 sentences with a total of 27858619 words.
        INFO:preprocess:Total: 95088 unique words in 1000000 sentences with a total of 27858619 words.
        INFO:preprocess:Creating dictionary of 30000 most common words, covering 99.5% of the text.
        INFO:preprocess:Saving to data/vocab.en.pkl.
        INFO:preprocess:Binarizing Ingles.tok.txt.
        INFO:preprocess:Saving to data/binarized_text.en.pkl.
        INFO:preprocess:Counting words in Espanol.tok.txt
        INFO:preprocess:140820 unique words in 1000000 sentences with a total of 28923828 words.
        INFO:preprocess:Total: 140820 unique words in 1000000 sentences with a total of 28923828 words.
        INFO:preprocess:Creating dictionary of 30000 most common words, covering 98.8% of the text.
        INFO:preprocess:Saving to data/vocab.es.pkl.
        INFO:preprocess:Binarizing Espanol.tok.txt.
        INFO:preprocess:Saving to data/binarized_text.es.pkl.
```

6.2.3 **Codificación del corpus**

Después de crear los diccionarios se creara un archivo pickle que contiene una lista de vectores numpy representando a cada línea en el parallel corpora, cada palabra en la línea está codificada con su id correspondiente en el diccionario. Estos archivos se convertirán en archivos HDF5, ya que este es un formato diseñado para almacenar grandes cantidades de datos, al igual que ser diseñado por un grupo sin ánimo de lucro. Posteriormente estos archivos se mezclan, aunque este último paso es opcional ya que puede ser costoso mezclar los dataset cada vez que se entrena un modelo. En la figura 12 se puede observar la codificación de las frases.

Figura 12. Codificación de corpus

```
In [3]: !python preprocess/convert-pkl2hdf5.py data/binarized_text.en.pkl data/binarized_text.en.h5
        !python preprocess/convert-pkl2hdf5.py data/binarized_text.es.pkl data/binarized_text.es.h5

        . . . . . 100000 . . . . . 200000 . . . . . 300000 . . . . . 400000 . . . . . .
        500000 . . . . . 600000 . . . . . 700000 . . . . . 800000 . . . . . 900000 . . . . . .
        . . 1000000 processed 1000000 phrases
        . . . . . 100000 . . . . . 200000 . . . . . 300000 . . . . . 400000 . . . . . .
        500000 . . . . . 600000 . . . . . 700000 . . . . . 800000 . . . . . 900000 . . . . . .
        . . 1000000 processed 1000000 phrases

In [4]: !python preprocess/shuffle-hdf5.py data/binarized_text.en.h5 data/binarized_text.es.h5 data/binarized_text.shuffled.en.h5

        Data len: 1000000
        Shuffled
        . . . . . 100000 . . . . . 200000 . . . . . 300000 . . . . . 400000 . . . . . 500
        000 . . . . . 600000 . . . . . 700000 . . . . . 800000 . . . . . 900000 . . . . . .
         1000000 processed 1000000 phrase pairs
```

6.3 **Diseño y entrenamiento del modelo**

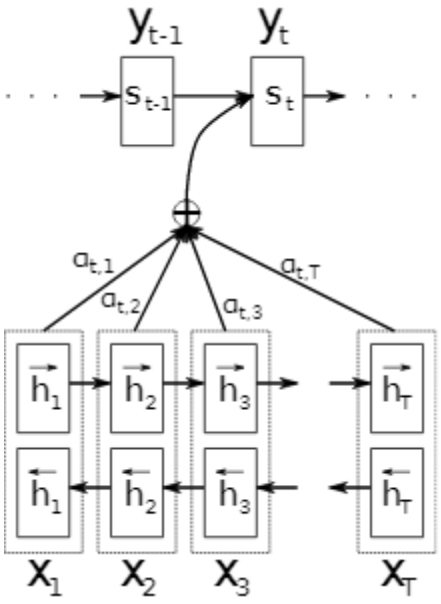
6.3.1 **Diseño del modelo**

Para este proyecto nos basamos en el traductor propuesto por Cho[12] el cual se puede ver en la figura 13, donde el codificador está compuesto por una RNN hacia adelante que lee la secuencia de entrada (x_1 a x_T) y calcula una secuencia de estados escondidos hacia adelante y otra RNN lee la secuencia en orden invertido de (x_T a x_1) generando una secuencia de estados escondidos hacia atrás. Cada RNN posee 1000 unidades escondidas GRU al igual que el decodificador.

Una anotación x_j es obtenida al concatenar el estado escondido hacia adelante y el estado escondido hacia atrás, esto contiene el resumen tanto de la palabra siguiente como la anterior.

En la arquitectura se usa una red multicapa con una sola capa escondido *maxout* para computar la probabilidad condicional de cada palabra objetivo. Se usa un algoritmo de gradiente estocástico (SGD por sus siglas en inglés) por mini lotes junto con Adadelta para entrenar cada modelo. Cada actualización de la dirección del SGD es computado usando un mini lote de 80 frases [12].

Figura13. Ilustración del modelo generando la palabra objetivo dado una frase fuente



Fuente: Neural machine translation by jointly learning to align and translate

6.3.2 **Diseño experimental**

Utilizamos la arquitectura antes expuesta con 1000 unidades GRU por cada RNN a la que llamamos arquitectura 1, utilizaremos 500 unidades GRU en la arquitectura 2.

La arquitectura 1 se entrenó con el *dataset* de Europarl y Europarl fracción, mientras que la arquitectura 2 se entrenó solo con el *dataset* Europarl. Estos textos de entrenamiento son de temas políticos y económicos.

El entrenamiento de la red genera 3 archivos explicados brevemente a continuación:

- Search_model.npz: archivo numpy que contiene los parámetros de la red
- Search_state.pkl: archivo pickle que contiene el estado de la red.
- Search_timing.npz: archivo numpy con estadísticas del entrenamiento.

6.3.3 **Infraestructura**

Como se dijo anteriormente Jupyter es una plataforma web la cual nos permite organizar nuestro trabajo sean códigos, textos o diferentes archivos, igualmente nos permite compartir avances de manera sencilla. Aun así la interacción de Jupyter con GUANE no es tan sencilla y depende de un equipo de trabajo que debe estar atento a su funcionamiento al igual que a su mantenimiento.

Figura 14. Infraestructura

En la figura 14 se muestra la forma de que un usuario pueda acceder a los recursos de GUANE debe ingresar a través de toc toc el cual es la compuerta de entrada la página de SC3 desde donde podemos acceder a GUANE el cual será el administrador de trabajos donde podemos llegar a jupyter, la herramienta que nos permite navegar de forma organizada el trabajo que hemos realizado.

6.4 **Evaluación del modelo**

Una vez entenados los modelos se usa el *dataset* news commentary como prueba para la traducción, el cual tiene una referencia confiable y por tanto podemos obtener un puntaje BLEU, hay que resaltar que el news commentary no está en ningún momento en la etapa de entrenamiento pero tiene un tema similar a Europarl al ser de economía y política. Con lo anterior se busca expandir la etapa de evaluación a textos que involucren estos temas, como los son las noticias locales y textos que no tengan que ver con estos temas como son cuentos o fragmentos de ellos, esto da a entender que el análisis de estos últimos resultados es cualitativo ya que no podemos obtener el puntaje BLEU sin una traducción de referencia.

6.4.1 **Resultados**

En la tabla 1 se muestran los resultados de los modelos realizados en este proyecto comparados con [12]. Esta comparación incluye los idiomas, el tamaño del *dataset* de entrenamiento, la arquitectura, el número de iteraciones, el tiempo y el puntaje BLEU obtenido.

Tabla 1. Comparación de resultados

Corpus	Idioma	Tamaño	Arquitectura	Iteraciones	Tiempo (horas)	BLEU
Europarl	ES-EN	54M	1	60000	806	27.60
Europarl	ES-EN	54M	2	30000	263	24.74
Europarl (fracción)	ES-EN	28M	1	45000	589	23.58
Cho[12]	EN-FR	348M	1		120*	28.45

Primeramente se nota que el número de iteraciones para cada modelo fue distinto, esto se debe a la complejidad computacional, ya que para aprovechar mejor los recursos no se podían entrenar más de 2 modelos simultáneamente y debido al tiempo para realizar este proyecto no se pudo extender estos tiempos de iteración. En [12] hay un tiempo de entrenamiento de alrededor 5 días.

Aun por lo dicho anteriormente se pudieron obtener resultados satisfactorios empezando por el modelo entrenado con Europarl y la arquitectura 1, donde su desempeño está por debajo del mejor puntaje BLEU de [12] por menos de un punto siendo que el tamaño del data set usado en este proyecto es una sexta parte del usada por Cho. Igualmente la fracción de Europarl de tan solo 28M de palabras se queda bastante corto en su puntaje, dando a entender que aunque es posible tener resultados coherentes con un *dataset* de este tamaño no serán los mejores y el tiempo usado extrapolado será muy parecido a Europarl dela arquitectura 1.

La arquitectura 2 pese a tener el menor tiempo de cómputo y un dataset de 54M obtuvo un puntaje BLEU superior a la arquitectura 1 con un *dataset* de 28M, mostrando la importancia del tamaño del *dataset*, ya que en todos los casos el diccionario era de 30000 palabras.

En la tabla 2 se pueden ver algunos ejemplos de las traducciones obtenidas por cada modelo realizado en el proyecto al igual que el traductor de google, estas frases se traducen del *dataset* de prueba News commentary. Hay que hacer énfasis en que NMT se desempeña bien en áreas similares a las de su entrenamiento y tiene menos flexibilidad que google.

Tabla 2. Muestra de traducciones

Frase español	**Europarl (arq1)**	**Europarl(arq2)**	**Europarl frac**	**Google**
No hablo de una Europa de dos velocidades.	I do not speak of a two-speed Europe.	I am not talking about a two-speed Europe .	I am not talking about a Europe of two speeds .	I am not talking about a two-speed Europe.
Ya existen precedentes.	There are precedents .	There are already precedents .	There are precedents.	There are already precedents.
En vez de ello, el foco inmediato debe mantenerse en impulsar la inversión y las exportaciones en economías con déficits en sus cuentas corrientes –como Francia, Italia y España (y el Reino Unido)– y estimular el consumo en economías con superávits, como Alemania y los Países Bajos.	Instead , the immediate focus must be maintained in boosting investment and exports in UNK economies , like France , Italy and Spain (and the United Kingdom) - and stimulating consumption in economies , like Germany and the Netherlands .	Instead , the immediate focus must be kept in promoting investment and exports in economies with deficits - such as France , Italy and Spain (and the UK - and stimulate the consumption in economies with UNK , like Germany and the Netherlands .	Instead , the immediate aim must continue to encourage investment and exports into economies in their own accounts - such as France , Italy and Spain (and the United Kingdom) - and stimulate consumption in UNK economies , as Germany and the Netherlands .	Instead, the immediate focus must be on boosting investment and exports in current account deficits - such as France, Italy and Spain (and the United Kingdom) - and stimulating consumption in surplus economies such as Germany And the Netherlands.
Se pueden ampliar esos precedentes.	They can be extended .	These precedents can be extended .	These precedents can be extended .	These precedents can be expanded.

Al comienzo de la guerra encabezada por EEUU en Irak, dos visiones contrapuestas daban forma a las predicciones acerca de los resultados.	At the beginning of the war led by the USA in Iraq , two opposing views were given to the forecasts of the results .	At the start of the war led by the United States in Iraq , two UNK visions will put a way to the predictions on results .	At the beginning of the war perpetrated by the United States in Iraq , two conflicting visions would give us a way to UNK the results .	At the beginning of the US-led war in Iraq, two opposing visions gave predictions about the results.
No se trata ni de una opción ni de un lujo: es una necesidad.	This is not an option or a luxury : it is a necessity .	It is neither an option nor a luxury : it is a necessity .	This is not a question of choice or luxury : it is a necessity .	It is neither an option nor a luxury: it is a necessity.
Estamos aceptando la carga que nos corresponde.	We are accepting the burden .	We are accepting the burden on us .	We are accepting the burden on us .	We are accepting the burden that corresponds to us.

Las traducciones de frases cortas tienden a ser más exactas en todos los modelos así como en google, los modelos no reconocen algunas palabras pero igual las alinean del modo correcto en la traducción manteniendo un sentido sintáctico en todas las frases.

En muchos casos las traducciones palabra a palabra no son correctas pero si su sentido, como el caso de la traducción de la frase de "al comienzo de la guerra encabezada por EEUU en Irak, dos visiones contrapuestas daban forma a las predicciones acerca de los resultados", la traducción dada por la arquitectura 1 de Europarl traduce "predicciones" como "*forecast*" que significa "pronóstico" y no como "*predictions*" como se esperaría, aun así esta traducción posee el mismo sentido de la frase fuente. Más ejemplos existen como lo son la traducción de "encabezados" por "*perpetrated*" o "comienzo" por "*start*".

Al aumentar el tamaño de la frase existen más probabilidades de encontrar palabras desconocidas (UNK) al igual que el rendimiento baja sobre todo en el Europarl entrenado con el *dataset* de 28 M de palabras, mostrando que un *dataset* de entrenamiento de al menos 54 M es necesario para un entrenamiento satisfactorio.

6.4.2 **Análisis de resultados cualitativos**

Como se expuso anteriormente esta sección consta de pruebas realizadas a diferentes textos de diferentes temas y su análisis cualitativo por la falta de una traducción de referencia para obtener puntaje BLEU. Las pruebas se realizaran con dos noticias de periódicos nacionales y la fracción de un cuento.

6.4.2.1 **Noticia 1**

La primera noticia[7] es un tema político y de mucha controversia en Colombia que fue el tratado de paz con las FARC, se utilizara solo una fracción de la noticia, a continuación se mostrara el texto en español usado para la traducción encerrado para poder diferenciarse:

"El hecho de que hayamos logrado firmar un acuerdo nos da la plena certeza de que estamos empezando a crear las condiciones para sentar las bases de una paz duradera y perdurable en una Colombia que lo está pidiendo a gritos", indica en una entrevista difundida hoy por la cadena francesa "France 24".

El representante de las Farc aseguró que su movimiento siempre ha aspirado a la paz y no va a escatimar esfuerzos para conquistarla, especialmente ahora que se han dado unas "condiciones excepcionales" para permitirlo.

El acuerdo definitivo de paz entre el Gobierno y las Farc se firmó el pasado 24 de noviembre al término de cuatro años de negociaciones en La Habana y después de que un primer texto alcanzado el 26 de septiembre fuera rechazado en las urnas el 2 de octubre.

Tomamos solo 3 párrafos de la noticia, ya que esto es solo para ver el desempeño de las redes ante esta situación, los resultados están dados en la tabla 3

Tabla3. Traducciones noticia 1

Europarl arquitecura 1	Europarl arquitectura 2	Europarl fracción
UNK the fact that we have succeeded in signing an agreement gives us the full certainty that we are beginning to create the conditions to lay the foundations of lasting peace and UNK in a Colombia which is calling for it , is indicated in an interview by the French company Mr noon . The representative of the UNK ensured that his movement has always brought to peace and it will not be spared any effort to	The fact that we managed to sign a compromise gives us full certainty that we are starting to create the foundations of a lasting peace and UNK in a Colombia that is called for UNK “ , which is indicated in an interview by the French UNK of 24 . (The representative of the UNK assured that their movement has always UNK to peace and will not be able to make any efforts to take , especially now that they have been given “	The fact that we have managed to sign a compromise gives us the full certainty that we are beginning to create the conditions for establishing a lasting peace base in Colombia , which is called for alarm “ , described in a interview with the French Government “ s UNK . The representative of UNK assured that his movement has always been given to peace and is not going to make any effort to prevent it , especially now that a few

[7] http://www.vanguardia.com/colombia/382692-timochenko-cree-que-el-acuerdo-sienta-las-bases-de-una-paz-duradera

save it , especially now that there have been “ exceptional conditions “ . The final agreement of peace between the government and the UNK was signed on 24 November at the end of four years of negotiations in Havana and then a first text reached 26 September was rejected by 26 October .	exceptional conditions “ . (The final peace agreement between the government and the UNK was signed last November at the end of four years of negotiations in Havana and after the first text reached 26 September was rejected by 2 October .	exceptional conditions have been put to it . The peace agreement between the Government and the UNK was signed on 24 November at the end of four years of negotiations in Cairo and then that a first text reached on 26 September was rejected on 2 October .

Aunque no existen muchas palabras desconocidas ha podido traducir en gran parte la fracción de documento manteniendo su estructura sintáctica, se puede ver que ha fallado los 3 modelos en “la cadena francesa “France 24” y en la arquitectura 1 de Europarl la traducción de 2 de Octubre a 26 de Octubre aunque mantiene el sentido de la oración un numero en este caso puede cambiar la información al usuario. Así mismo se puede notar que la palabra “FARC” es traducida al carácter especial UNK, lo cual es de esperarse al ser una sigla que no es usado en el *dataset* de entrenamiento.

6.4.2.2 **Noticia 2**

La segunda noticia[8] es una noticia económica y será mostrada a continuación:

En la jornada del jueves, el peso colombiano se apreció 0,32 % y fue una de las monedas de mejor desempeño entre las economías emergentes. "Lo anterior se dio a pesar de que los precios internacionales del petróleo presentaron una corrección a la baja", señala la firma Credicorp Capital. En la jornada de hoy, el dólar se negocia a la baja y cae por debajo de los 3.000 pesos. Por un billete verde se pagan en promedio $ 2.993, es decir, la divisa pierde casi 7 pesos frente a la TRM del día, $ 3.000,47. En el mercado accionario, el índice Colcap no presenta cambios importantes. Retrocede levemente un 0,01 por ciento. La acción de PFAVH gana 4,61 por ciento. Los papeles de Fabricato pierden 3,70 por ciento.

Igualmente mostraremos en la tabla 4 los resultados de las traducciones.

[8] http://www.eltiempo.com/economia/indicadores/dolar-en-colombia/16774484

Tabla 4. Traducciones noticia 2

Europarl arquitecura 1	Europarl arquitectura 2	Europarl fracción
In the course of the vote on Thursday , there was UNK UNK UNK and it was one of the oldest currencies of the emerging economies .	In the course of Thursday , the UNK is UNK % and was one of the UNK coins between the emerging economies .	In the course of Thursday , the UNK burden UNK UNK % and was one of the UNK currencies between emerging economies .
The previous speaker was in spite of the fact that international oil prices did have a correction to leave it , " UNK UNK".	One earlier was taken out of the fact that international oil prices came to an end to the bottom " , he points out the UNK .	What happened to the fact that international oil prices gave a correction to the bottom " , it is a question of UNK UNK .
In the course of the day , the dollar is negotiated in the short and fall below the three quarters , and a green card is paid on average UNK UNK , which is to say , the currency loses almost 7 UNK on the agenda .	In the days of the day , the dollar is starting to the low , and it is not a matter of EUR UNK UNK , it is said , the currency loses almost 7 UNK to the UNK , UNK UNK .	In today " s day , the dollar is broken down to the dollar and falls below the UNK , which is why a Green Card is paid on average UNK , that is to say the currency loses almost 7 000 out of the UNK , UNK UNK .
In the UNK market , the UNK rate does not present any major changes .	In the UNK market , the UNK rate does not produce significant changes . UNK UNK one per cent .	In the UNK market , the UNK rate does not produce important changes , UNK UNK per cent .
The action of UNK UNK per cent .	The action of UNK is UNK per cent .	UNK action by UNK % .
The UNK of UNK lose UNK per cent .	The UNK were UNK per cent .	UNK UNK UNK UNK .

Ciertamente se observa que existen más palabras desconocidas en este tipo de texto sobre todo por siglas y nombres de entidades y aunque los modelos intentan traducir y mantener un sentido, pierde coherencia en frases largas o donde existen muchas palabras desconocidas.

6.4.2.3 **Fracción cuento**

Por último se probara los modelos con una fracción de un cuento de los hermanos Grimm[9] llamado las tres hojas de la serpiente. A continuación se mostrara la fracción escogida en español

[9] http://www.grimmstories.com/es/grimm_cuentos/las_tres_hojas_de_la_serpiente

Vivía una vez un hombre tan pobre, que pasaba apuros para alimentar a su único hijo. Díjole entonces éste:

- Padre mío, estáis muy necesitado, y soy una carga para vos. Mejor será que me marche a buscar el modo de ganarme el pan.

Dióle el padre su bendición y se despidió de él con honda tristeza.

Sucedió que por aquellos días el Rey sostenía una guerra con un imperio muy poderoso. El joven se alistó en su ejército y partió para la guerra. Apenas llegado al campo de batalla, se trabó un combate. El peligro era grande, y llovían muchas balas; el mozo veía caer a sus camaradas de todos lados, y, al sucumbir también el general, los demás se dispusieron a emprender la fuga. Adelantóse él entonces, los animó diciendo:

- ¡No vamos a permitir que se hunda nuestra patria!

A continuación se muestran las traducciones hechas por los modelos.

Tabla 5. Traducciones fracción cuento

Europarl arquitecura 1	Europarl arquitectura 2	Europarl fracción
This is a very poor man , which has trouble to feed his only child . - my fellow Members , Mr UNK , have been very committed , and I am a burden on the market . I admire his blessing and he was disappointed with deep sadness . Anyone who said the King UNK a war with a very powerful empire , the young people were involved in his army and went to the fight against the whole of the world , the danger was large , and we will remember that there is a lot of other than the other . Let us not allow our homeland to collapse !	UNK once a poor man , who were UNK to feed his single child , will then read that : - Mr Chatzimarkakis , I will be very much needed , and I am a burden on it - it would be very much to look to me to seek the way of UNK . I would like to apologise for the father , and he will be told him with great sadness . I would argue that , for those days , the young people were UNK in their army , and I would like to make many of you - the UNK of all sides , and I would like to stress the others would like to take the UNK , and I would like to thank you : - We are not going to allow our homeland to be lifted !	We are once again a very poor man , who made a lot of trouble to feed his only child , who is saying : Ladies and gentlemen , I am very committed to you , and I am a UNK , and I hope that I will be able to look at the way of UNK the bread . It is the father ‘ s father ‘ s father ‘ s name , with deep sadness . Having said that , for example , the bombing of a war with a very powerful empire ‘ s visit to the fight against terrorism is a major one , and I will use it every day - because of course the rest of all of us - and the rest of the rest , is that others are going to take a stop to the UNK . We are not going to allow

		our own homeland to happen .

El cuento por su narrativa es muy diferente a los textos de entrenamiento y aunque no hay tantas palabras desconocidas como en la noticia 2 si hay ciertas incoherencias como lo es la traducción de la frase “Padre mío, estáis muy necesitado, y soy una carga para vos”, el cual ningún modelo pudo traducir y deja de evidencia que la fortaleza de la traducción del NMT es en base a textos de áreas específicas relacionadas al entrenamiento.

7. CONCLUSIONES

- Se estima que con un *parallel corpus* de 54 M de palabras para el entrenamiento y la arquitectura 1 se puede obtener un buen desempeño.

- La arquitectura 2 usando 500 unidades GRU por cada RNN en el modelo tiene resultados muy buenos y con un tiempo de cómputo menor a los modelos que usan 1000 unidades GRU, este modelo puede ser usado para comprobar hipótesis debido a la similitud de los datos con la arquitectura 1.

- A medida que la longitud de la frase aumenta el desempeño de los modelos decrece, esto sucede incluso con el traductor de google como se ve en la tabla2.

- Después de entrenados los modelos estos aprenden a alinear las palabras en fin de dar un sentido a la frase y aunque no traducen las palabras una a una, muchas traducciones tienen sinónimos y mantienen el sentido de lo dicho.

- El desempeño de los NMT incrementa al ser usado en la traducción de textos con naturaleza similar al corpus de entrenamiento.

8. RECOMENDACIONES

- Para futuras pruebas revisar la arquitectura 2 con el mismo número de iteraciones para ser comparado con la arquitectura 1.

- Al momento de realizar los planes se deben tener en cuenta los tiempos de cómputo obtenidos en este proyecto para poder llevar cualquier trabajo a un término fijo.

- Un *dataset* más grande podría mejorar la puntuación BLEU al igual que el análisis de textos muy diferentes a los entrenados en el modelo.

- Evaluar los modelos con textos de la misma naturaleza del corpus de entrenamiento.

9. **BIBLIOGRAFIA**

[1] BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv, 2014.

[2] BERGER, Adam L., et al. The Candide system for machine translation. En Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, 1994.

[3] BLUNSOM, Phil; KALCHBRENNER, Nal. Recurrent continuous translation models. En Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.

[4] CHO, Kyunghyun, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv, 2014.

[5] CHO, Kyunghyun, et al. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv, 2014.

[6] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. Neural computation, 1997.

[7] HOCHREITER, Sepp. Untersuchungen zu dynamischen neuronalen Netzen.. Tesis Doctoral. diploma thesis, institut für informatik, lehrstuhl prof. brauer, technische universität münchen,1991.

[8] LIPTON, Zachary C.; BERKOWITZ, John; ELKAN, Charles. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv, 2015.

[9] LUONG, Minh-Thang; PHAM, Hieu; MANNING, Christopher D. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv, 2015.

[10] PAPINENI, Kishore, et al. BLEU: a method for automatic evaluation of machine translation. En Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

[11] ROBERT, Christian. Introduction. En: Machine Learning A Probabilistic Perspective, 2014.

[12] SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. En Advances in neural information processing systems, 2014.