UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

# Generación de textos en ruso mediante técnicas de Aprendizaje Automático para la industria del lenguaje

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática

*Autor:* Mykyta Grygoryev

*Tutor:* Francisco Casacuberta Nolla
Mercedes García-Martínez

Curso 2021-2022

# Resum

Hui dia, els avanços en l'àrea del Processament del Llenguatge Natural i l'Aprenentatge Automàtic permeten l'anàlisi, la comprensió i la generació automàtica de text cada vegada més precís i fluid. L'objectiu d'aquest treball final de grau és la creació automàtica d'exemples de text en rus a partir de dades de text ja existents mitjançant tècniques d'aprenentatge automàtic. S'han emprat xarxes neuronals i recursos lingüístics per a la generació automàtica de text en rus. Per al desenvolupament del treball s'han utilitzat dades de domini públic. El sistema genera nous textos utilitzant informació d'*embeddings* entrenades amb una ingent quantitat de dades en models de llenguatge neuronals. La generació d'aquests textos incrementa el corpus utilitzat a l'entrenament de models per a tasques de Processament del Llenguatge Natural com ara la traducció automàtica. També podria aplicar-se a d'altres tasques com, per exemple, la generació de resums automàtics o als parafrasejadors de textos. Finalment, s'ha realitzat una anàlisi dels resultats obtinguts mitjançant l'avaluació de la qualitat dels textos generats, els quals s'han afegit a l'entrenament de models de traducció automàtica neuronal. Aquests models s'han comparat realitzant, d'una banda, una anàlisi quantitativa amb la comparació dels diferents mètodes mitjançant diverses mètriques automàtiques típiques utilitzades en traducció automàtica, així com el mesurament dels temps emprats i la quantitat de text generat per un bon ús en la indústria del llenguatge i, d'altra banda, una anàlisi qualitativa, on s'han exposat exemples de traducció generats pels models de traducció entrenats i s'han comparat entre ells.

**Paraules clau:** Processament del Llenguatge Natural, augment de dades, xarxes neuronals, grans conjunts de dades, Aprenentatge Automàtic, Intel·ligència Artificial, Aprenentatge Profund.

# Resumen

Hoy en día los avances en el área del Procesamiento del Lenguaje Natural y el Aprendizaje Automático permiten el análisis, la comprensión y la generación de texto automáticamente cada vez más precisa y fluida. El objetivo de este trabajo final de grado es la creación automática de ejemplos de texto en ruso, a partir de datos de texto ya existentes mediante técnicas de aprendizaje automático. Se han empleado redes neuronales y recursos lingüísticos para la generación automática de texto en ruso.

Para el desarrollo del trabajo se han utilizado datos de dominio público. El sistema genera nuevos textos utilizando información de *embeddings* entrenadas con una ingente cantidad de datos en modelos de lenguaje neuronales. La generación de dichos textos incrementa el corpus utilizado para el entrenamiento de modelos para tareas del Procesamiento del Lenguaje Natural como la traducción automática. También podría aplicarse a otras tareas como la generación de resúmenes automáticos o parafraseadores de textos.

Por último, se ha realizado un análisis de los resultados obtenidos evaluando la calidad de los textos generados y se han añadido al entrenamiento de modelos de traducción automática neuronal. Estos modelos se han comparado realizando un análisis cuantitativo, comparando los distintos métodos mediante varias métricas automáticas típicas utilizadas en traducción automática y se han medido los tiempos empleados y la cantidad de texto generado para un buen uso en la industria del lenguaje, y un análisis cualitativo,

donde se han expuesto ejemplos de traducción generados por los modelos de traducción entrenados y se han comparado entre sí.

**Palabras clave:** Procesamiento del Lenguaje Natural, aumento de datos, redes neuronales, grandes conjuntos de datos, Aprendizaje Automático, Inteligencia Artificial, Aprendizaje Profundo.

# Abstract

Current progress in the areas of Natural Language Processing and Machine Learning allows for the analysis, understanding and automatic generation of increasingly accurate and fluid text. The objective of this final degree project is automatically creating text examples in Russian from existing text data using machine learning techniques. Neural networks and linguistic resources have been used for the automatic generation of text in Russian. To develop this project, data from the public domain have been used. The system generates new texts using information from embeddings trained with a huge amount of data in neural language models. The generation of these texts increases the corpus used to train models for several Natural Language Processing tasks, for instance, machine translation. It could also be applied to other tasks such as generating automatic summaries or to text paraphrasers. Finally, an analysis of the results obtained evaluating the quality of generated texts has been carried out and those texts have been added to the training process of neural machine translation models. On the one hand, these models have been compared by performing a quantitative analysis, comparing the different methods by means of several typical automatic metrics used in machine translation and measuring the times spent and the amount of text generated for good use in the language industry. On the other hand, they have been compared through a qualitative analysis, where examples of translation generated by the trained translation models have been exposed and compared with each other.

**Key words:** Natural Language Processing, neural networks, Big Data, Machine Learning, Artificial Intelligence, Deep Learning.

# Índice general

# Índice de figuras



# Índice de tablas

# CAPÍTULO 1
# Introducción

Este capítulo introductorio brinda al lector un contexto general para enmarcar el trabajo presentado en este documento. A continuación, describimos la motivación que nos llevó a realizar este trabajo y los objetivos propuestos del mismo. Además, se hace una introducción al campo de la MT (*Machine Translation*) y a la importancia del DA (*Data Augmentation*).

Por último, se resume la estructura del resto del documento, permitiendo al lector tener una visión general del trabajo realizado.

## 1.1 Motivación

La utilización de conjuntos de datos grandes y de calidad para entrenar modelos para tareas especificas del procesamiento del lenguaje natural o en inglés *Natural Language Processing* (NLP) como traducción automática o en inglés *Machine Translation* (MT), generación automática de resúmenes, parafraseo, generación de texto o sistemas de diálogo, es esencial para conseguir buenos resultados en dichas tareas. Sin embargo, no siempre se dispone de tales cantidades de datos o la calidad de los mismos no es significativamente buena para su uso durante el entrenamiento de un modelo de lenguaje. Por ello, el interés en el aumento de datos de forma artificial se ha visto incrementado recientemente en el campo del NLP, debido a un mayor desarrollo en dominios de bajos recursos, nuevas tareas y la popularidad de las redes neuronales a gran escala, las cuales requieren de grandes cantidades de datos de entrenamiento. El aumento de datos en el entorno de NLP supone un desafío por la naturaleza discreta de los lenguajes y, sobre todo en la tarea de MT, debido a las distintas y diversas características que pueden llegar a tener los idiomas.

## 1.2 Objetivos

El objetivo principal de este proyecto consiste en el estudio y la experimentación de técnicas de aumento de datos a nivel de palabra y su influencia en modelos de traducción automática neuronal. El trabajo se divide en tres objetivos:

- Generación de texto natural en ruso: creación de nuevos ejemplos de datos para la tarea de traducción automática utilizando un modelo de lenguaje pre-entrenado.
- Comparativa de técnicas de aumento de datos en modelos de traducción automática: análisis cuantitativo y cualitativo de las traducciones de los modelos de traducción automática entrenados con datos aumentados.

- Mejora de la calidad de las traducciones generadas por los modelos de traducción automática utilizando los datos aumentados automáticamente.

## 1.3 Traducción Automática

La traducción automática es un problema clásico en NLP. La resolución de este problema consiste en que dada una unidad de texto como puede ser un documento, un párrafo o una simple frase en un idioma dado, se consiga su texto traducido al idioma objetivo.

En un mundo tan globalizado como en el que estamos viviendo ahora mismo, es de vital importancia que la inmensa cantidad de texto que se encuentra constantemente a nuestro alrededor y que observamos a diario en páginas *web*, correos electrónicos, redes sociales, periódicos y otros medios de comunicación, pueda ser entregada y entendida por la máxima cantidad de personas sin importar los idiomas que conozca. Todo ello supone un gran reto a la hora de conseguir dichas traducciones, empleando la menor cantidad de recursos y tiempo posibles.

Los primeros sistemas de traducción automática están basados en reglas [41]. Estos sistemas de traducción automática se crean manualmente a partir de expertos. Aunque las reglas quedan definidas de forma sólida gracias a los expertos, el proceso en si es costoso y no generaliza bien a todos los dominios e idiomas existentes.

Con el paso del tiempo, los sistemas basados en reglas se fueron sustituyendo por los modelos de aprendizaje automático clásico. En primer lugar la traducción automática estadística o por sus siglas en inglés *Statistical Machine Translation* (SMT) [4], la cual emplea conjuntos de datos paralelos para crear modelos de forma automática mediante técnicas de estadística. Debido a esto, las reglas creadas por expertos ya no eran necesarias puesto que el sistema en si encontraba los patrones ocultos e iba creando las reglas en base al conjunto de datos empleados. Por ende, este sistema era mucho más rápido y menos costoso que el basado en reglas pero se sacrifica la supervisión humana.

En los últimos años, el aprendizaje automático profundo está revolucionando la forma en que se construyen los sistemas de traducción automática. El auge del aprendizaje automático profundo ha hecho que la mayoría de los modelos que se emplean actualmente en traducción automática estén basados en redes neuronales artificiales. Por ello, la traducción automática neuronal o *Neural Machine Translation* (NMT) [43], está teniendo un crecimiento exponencial gracias al desarrollo de nuevas unidades de procesamiento gráfico o *Graphics Processing Unit* (GPU) que tienen, cada vez, una mayor capacidad de cómputo, lo cual permite ejecutar varias operaciones en paralelo lo que, a su vez, hace posible la NMT. Actualmente, la NMT está basada en un modelo secuencia a secuencia. Se han empleado redes neuronales recurrentes o *Recurrent Neuronal Network* (RNN) [8] (ver 2.2.1), una para la codificación de la frase en un vector mediante un codificador y, otra para la decodificación del vector en una frase mediante un decodificador.

Con el paso del tiempo, aparecieron nuevos modelos que emplean mecanismos de atención, los cuales permiten enfocarse en diferentes palabras de la frase en base a la relación de las palabras en la frase fuente y destino, lo cual permitió generar unas representaciones más discriminatorias que permitieron obtener un rendimiento más alto. Actualmente, el estado del arte en traducción automática está basado en el modelo *Transformer* [45]. Este modelo permite paralelizar secuencias utilizando mecanismos de atención evitando cuellos de botella debido a las RNN, siendo computacionalmente más eficiente. Inicialmente, este modelo se ideó para traducción automática, entrenándolo con enormes cantidades de texto para lograr el mayor rendimiento visto hasta la fecha en la tarea

de traducción automática. Posteriormente, se fueron refinando sus pesos para adoptarlo, con el fin de usarse en otras tareas de NLP.

## 1.4 Aumento de Datos

Se podría definir la presente década como la "Época del *Big Data*" por la masiva cantidad de datos de texto, voz e imagen que se generan a diario. Sin embargo, la calidad de dichos datos no siempre es buena, llegando a ser incluso insuficiente e irrelevante. Además, en algunos casos, como por ejemplo en el caso de los idiomas con bajos recursos lingüísticos, no hay suficiente cantidad de datos para poder realizar entrenamientos de modelos de inteligencia artificial, en concreto, en NLP.

A lo largo de los últimos años, fueron apareciendo diversas técnicas de aumento de datos con el objetivo de generar nuevos datos de calidad. Uno de los campos de inteligencia artificial que más se beneficia es el campo de la visión por computador, en concreto, las tareas de clasificación de imágenes, detección de objetos, y segmentación de imágenes. Entre las diversas técnicas que se emplean en visión por computador, cabe destacar el recorte, cambio de tamaño, volteo, rotación, cambio en el brillo y el contraste [29]. Además, hoy en día se emplean técnicas como la conversión de la imagen a blanco y negro o la aplicación de desenfoque. En la tarea de clasificación de imágenes, se observó que al emplear dichas técnicas se han obtenido mejoras significativas en la tasa de acierto.

En el campo de NLP, existen diversas y muy notables técnicas de aumento de datos. En concreto, los modelos de traducción automática precisan de enormes cantidades de datos que, a su vez, requieren de un alto nivel de calidad para su correcto entrenamiento. Además, debido al enorme impacto socio-cultural que estamos experimentando en los últimos años, cada vez se hace más necesario disponer de datos más variados que engloben la mayor cantidad de idiomas y dominios posibles, con el fin de conseguir mejores resultados en las diversas tareas de NLP. Las técnicas más populares de aumento de datos en NLP son: *Easy Data Augmentation* (EDA) [47] (ver 2.4.3), *Back-translation* (ver 2.4.1) [14] y el empleo de modelos de lenguaje pre-entrenados (ver 2.4.2), como BERT [10] o GPT-2 [32].

## 1.5 Estructura de la memoria

El presente trabajo está constituido por un total de seis capítulos. A continuación, se describen los contenidos tratados en cada capítulo:

En el **capítulo 1**, se realiza una introducción del tema que se aborda a lo largo del documento, además se detalla la motivación y los objetivos del trabajo fin de grado.

En el **capítulo 2**, se expone de forma teórica las diferentes aproximaciones de la traducción automática existentes y sus características principales. Además, se presentan las distintas técnicas de aumento de datos empleadas en NLP.

En el **capítulo 3**, se presenta al lector las distintas técnicas de aumento de datos que se han empleado en el presente proyecto para la generación de nuevos conjuntos de datos y, las características particulares de cada método y como se han resuelto los retos que se han planteado a la hora de generar nuevos textos en ruso.

En el **capítulo 4**, se muestra al lector el marco experimental del proyecto, el cual engloba las tecnologías que han sido empleadas en el desarrollo del trabajo, la forma de evaluación de los modelos de traducción entrenados, la preparación de los datos inicia-

les, la creación de los distintos conjuntos de datos generados, los experimentos que se han realizado y el hardware disponible en el entrenamiento de dichos modelos.

En el **capítulo 5**, se realiza el análisis cualitativo y cuantitativo de los diversos modelos de traducción automática entrenados a lo largo del proyecto.

En el **capítulo 6**, se presenta las conclusiones obtenidas a partir de la experimentación realizada y se exponen diferentes caminos a tomar en próximos trabajos relacionados con el aumento de datos en traducción automática neuronal.

Asimismo, a estos seis capítulos se referencia la bibliografía que se ha consultado en la realización del trabajo.

# CAPÍTULO 2
# Estado del arte

En el presente capítulo, se presenta al lector la historia de las tecnologías más relevantes en el campo de la traducción automática. Además, se presentan diversas técnicas que se emplean en el aumento de datos.

## 2.1 Traducción Automática Estadística

Los sistemas de traducción automática estadística son un paradigma de traducción automática donde las traducciones son generadas a partir de modelos estadísticos, cuyos parámetros derivan del análisis de un conjunto de datos paralelos compuestos por un texto fuente y un texto destino.

Así pues, la SMT se puede entender como la búsqueda de la hipótesis de traducción más probable dada una frase de origen. Por lo que, dada una frase de entrada en un idioma de entrada, el objetivo del sistema de traducción estadística es, desde una perspectiva formal, encontrar su correspondiente traducción en el idioma de salida. Dicho objetivo, es formalizado como sigue [5]:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) \tag{2.1}$$

de la ecuación anterior se deduce que, el objetivo de una SMT es encontrar la secuencia de salida más probable ($\hat{\mathbf{y}}$) dada una frase de entrada ($\mathbf{x}$). Dicha ecuación define el decodificador de una SMT.

No obstante, estos modelos suelen estar combinados con el modelo log-linear para el parámetro $p(\mathbf{y}|\mathbf{x})$, por lo que el problema se modela de la siguiente manera [23]:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \left\{ \sum_{\mathrm{n}=1}^{N} \lambda_n \log\left(h_n\left(\mathbf{x}, \mathbf{y}\right)\right) \right\} \tag{2.2}$$

donde $N$ es la cantidad de características, $h_n\left(\mathbf{x}, \mathbf{y}\right)$ define la $n$-ésima función que representa cada una de las características que son relevantes para la traducción, como por ejemplo la selección de plantillas de alineación, alineación a nivel de n-gramas o penalización a nivel de palabra o n-grama [27], y, por último, $\lambda_n$ representa los pesos asignados a la combinación log-linear para cada $n$-ésima característica.

## 2.2 Traducción Automática Neuronal

Las técnicas de aprendizaje automático profundo están reinventando la forma en la cual se desarrollan sistemas de traducción automática en la actualidad, no solo en la investigación académica sino también en su uso industrial. En los últimos años, la traducción automática neuronal, mediante el uso de redes neuronales, ha logrado resultados de vanguardia en múltiples pares de idiomas convirtiéndose así en la principal herramienta para abordar el desafío de la traducción automática [43].

La traducción automática neuronal tiene por objetivo estimar una distribución condicional desconocida que denominaremos $P(\mathbf{y}|\mathbf{x})$ y dado un conjunto de datos $D$, en el cual $\mathbf{x}$ e $\mathbf{y}$ son las variables aleatorias que representan las frases de entrada y salida. En cuanto a la manera de solucionar el problema de la traducción, la traducción automática neuronal puede resolverlo a nivel de documento, párrafo o frase.

En la traducción a nivel de frase, asumiendo una frase de entrada $\mathbf{x} = (x_1, \ldots, x_S)$ en el que $S$ se corresponde con el número total de palabras de entrada, y una frase de salida $\mathbf{y} = (y_1, \ldots, y_T)$, en el que $T$ comprende el número total de palabras de salida, empleando la regla de la cadena, la distribución condicional puede describirse como:

$$\hat{\mathbf{y}}_1^{\hat{T}} = \arg\max_{T, \mathbf{y}_1^T} \prod_{t=1}^{T} Pr_\theta(y_t | y_1^{t-1}, c(x_1^S)) \tag{2.3}$$

donde $y_t$ representa la palabra traducida actual, que es generada a partir de las palabras anteriores $y_1^{t-1}$ que han sido traducidas previamente con un tipo de representación denotado por la función $c$ de la frase de entrada $\mathbf{x}_1^S$, y empleando los parámetros del modelo $\theta$.

Actualmente, la inmensa mayoría de modelos de traducción automática neuronal utilizan el modelo codificador-decodificador en conjunto con arquitecturas neuronales profundas [6].

### 2.2.1. Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNN) incorporan la retroalimentación entre neuronas por lo que se consigue crear temporalidad, permitiendo que la red tenga memoria. Por ello, son empleadas para el desarrollo de codificadores y decodificadores potentes, convirtiéndose en el enfoque estándar y efectivo.

En la figura 2.1, se puede observar un ejemplo de una red neuronal recurrente. Así, una red neuronal recurrente es un tipo de red neuronal que está compuesta por un estado oculto denotado por $\mathbf{h}$ y una salida opcional $\mathbf{y}$ que opera en una secuencia de longitud variable $\mathbf{x}$. Dicho estado oculto, es actualizado para cada periodo de tiempo $t$ de la siguiente manera:

$$h_t = f_h(x_t, h_{t-1}) \tag{2.4}$$

$$y_t = f_o(h_t) \tag{2.5}$$

donde $h_t$ y $x_t$ son, respectivamente, el estado oculto de la red neuronal y el elemento perteneciente a la secuencia de longitud variable $\mathbf{x}$ en el instante $t$, y $f_h$ y $f_o$ son las funciones de activación de las neuronas ocultas y de las de salida respectivamente. La RNN es capaz de aprender la distribución de probabilidad sobre una secuencia de longitud variable

**x** si es entrenada para predecir el siguiente elemento de una secuencia. La salida en cada instante $t$ da como resultado la distribución condicional $p(x_t|x_1,\ldots,x_{t-1})$.

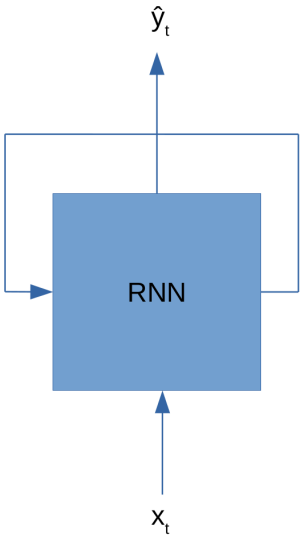


**Figura 2.1:** Arquitectura de una RNN simple

### 2.2.2. Arquitectura Codificador-Decodificador

Un codificador-decodificador, representado en la figura 2.2, emplea dos redes neuronales recurrentes de las cuales una codifica una secuencia de longitud variable en una representación vectorial de longitud fija, y la otra decodifica una representación vectorial en otra secuencia de elementos [6]. Por lo tanto, dichas redes neuronales se entrenan con el fin de maximizar la probabilidad logarítmica condicional dado un conjunto de pares de entrenamiento definidos como $D = \{(x_1, y_1), \ldots, (x_T, y_T)\}$, cuya definición formal sería:

$$\arg\max \sum_{t=1}^{T} \log p\left(y_t|y_{<t}, \mathbf{x}\right) \tag{2.6}$$

donde **x** es la frase de entrada, $y_t$ es la palabra de salida en el instante $t$ y $y_{<t}$ son todas las palabras anteriores de salida generadas hasta el instante $t$. Estas redes neuronales tiene como entrada la frase en el lenguaje fuente u origen, y su salida es la frase que contiene la traducción en el lenguaje destino u objetivo. Además, cabe destacar que el enfoque codificador-decodificador consiste en cuatro componentes básicos, las capas de *embedding* [16], las redes neuronales de codificación y las de decodificación y una capa de clasificación [43].

El objetivo de la capa de *embedding* es hacer representación continua de la frase de entrada. Por lo tanto, esta capa mapea un vector de elementos discretos $\mathbf{z}_t$ en un vector continuo $\mathbf{z}_t \in \mathbb{R}^d$, en el que la $d$ indica el tamaño del vector.

#### Codificador

Las redes neuronales recurrentes que se utilizan normalmente en los codificadores son del tipo LSTM (*Long Short-Term Memory*) [18] o GRU (*Gated Recurrent Unit*) [11]. Por lo tanto, la red codificadora se encarga de mapear los *embeddings* de entrada en representaciones continuas ocultas. Asimismo, con el objetivo de que el codificador aprenda las representaciones expresivas, tiene que ser capaz de modelar el orden y las dependencias

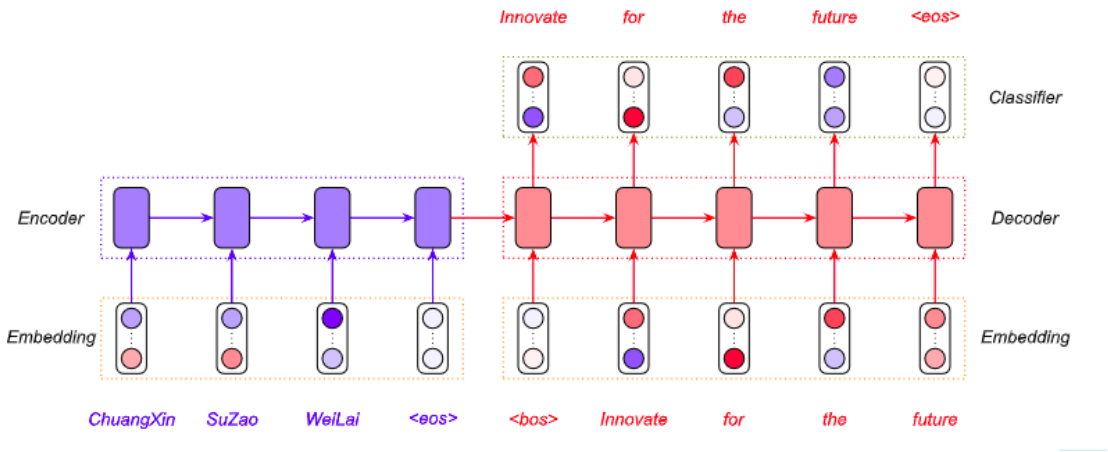


**Figura 2.2:** Arquitectura codificador-decodificador con dos capas de *embedding*, una red de codificadores, una red de decodificadores y una capa de clasificación, empleada en traducción automática neuronal [43].

que existen en el idioma de la frase de entrada. Asumiendo que los codificadores están representados por una RNN, estos leen cada elemento $x_t$ de una secuencia de entrada, y el estado oculto, $h_t$ de la RNN se define en la ecuación 2.7. El estado oculto $h_S$ es un resumen de toda la secuencia de entrada [40].

$$h_t = COD(x_t, h_{t-1}) \tag{2.7}$$

$$\mathbf{c} = q(h_1, \ldots, h_S) \tag{2.8}$$

donde $x_t$ es una secuencia de entrada al codificador, $h_t \in \mathbb{R}^t$ es el estado oculto en el instante $t$.

La ecuación 2.8 define $\mathbf{c}$ que se trata de un vector generado en base a la secuencia de estados ocultos que tiene una función no lineal $q$.

De este modo, aplicando iterativamente la función de transición de estado $COD$ (ver ecuación 2.7) sobre la frase de entrada, se puede emplear el estado final $h_S$, siendo $S$ el último elemento de la frase de entrada, como la representación de toda la frase y utilizarla para insertarla en el decodificador. Además, $\mathbf{c}$ es el vector de contexto generado en base a la secuencia de estados ocultos, mientras que $q$ es una función no lineal.

### Decodificador

El objetivo del decodificador es generar el texto de salida $\mathbf{y}$ dando como entrada las palabras predichas anteriormente y el vector de contexto $\mathbf{c}$. Por lo tanto, dado el elemento de comienzo $y_0 = < bos >$ y el estado inicial $s_0 = h_S$, el decodificador, representado como $DEC$ en la fórmulas, comprime la historia decodificada en un vector de estados $\mathbf{s}_t \in \mathbb{R}^d$, descrito formalmente queda de la siguiente manera:

$$\mathbf{s}_t = DEC(y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}) \tag{2.9}$$

**Capa de clasificación**

La capa de clasificación se encarga de predecir la distribución de los tokens de salida. Así, La capa de clasificación normalmente es una capa lineal que contiene la función de activación *softmax*. Por ello, suponiendo que el vocabulario del idioma de salida es $V$, y $|V|$ es el tamaño del vocabulario, dada la salida del decodificador $\mathbf{s}_t \in \mathbb{R}^d$, la capa de clasificación mapea $\mathbf{h}$ a un vector $\mathbf{z}$ en el espacio del vocabulario $\mathbb{R}^{|V|}$ con un mapa lineal. Tras el mapeo, se aplica la función *softmax* para asegurar que el vector es una probabilidad válida:

$$softmax\,(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^{|V|} \exp(z_k)} \tag{2.10}$$

donde se utiliza $z_i$ para exponer el i-ésimo componente del vector $\mathbf{z}$.

### 2.2.3. Mecanismo de atención

El mecanismo de atención considera las asociaciones entre cada palabra de entrada y cualquier palabra de salida y la usa para crear una representación vectorial de toda la secuencia de entrada. Por un lado, la autoatención consiste en extender el mecanismo de atención en el codificador. Por lo tanto, dicha extensión consiste en el cómputo de la asociación entre cualquier palabra de entrada y cualquier otra palabra de la entrada, en vez de calcular la asociación entre una palabra de entrada y una de salida. Por ello, los mecanismos de autoatención se han convertido en una parte integral del modelado de secuencias y modelos de traducción convincentes en varias tareas, lo que permite modelar las dependencias sin tener en cuenta su distancia en las secuencias de entrada o salida [2, 21].

### 2.2.4. Transformer

El *Transformer* [45] es una nueva arquitectura en la cual se reemplazan las capas de RNN, tanto en el codificador como en el decodificador, por redes neuronales basadas en la autoatención, además de varios cambios adicionales de la arquitectura tradicional de codificador-decodificador.

El codificador y decodificador están compuestos por la unión de varias capas de autoatención, las cuales forman bloques en la parte superior de cada componente, y redes neuronales no recurrentes.

Por un lado, el codificador está compuesto por la unión de seis bloques idénticos situados en la parte superior del codificador, donde cada bloque está compuesto por dos capas. La primera capa, esta compuesta por mecanismo multicabezal de autoatención, y la segunda es una simple red completamente conectada *feed-forward*. Además, se emplea una conexión residual sobre cada una de las dos capas, seguida de una capa de normalización.

Por otro lado, el decodificador está formado por la unión de 6 bloques idénticos, los cuales guardan una semejanza con la estructura de los bloques del codificador, pero a diferencia de estos últimos, se añade una capa multicabezal de autoatención en la que el valor y las claves pertenecen a la salida del codificador y la consulta, a la salida de la primera capa de atención del decodificador. Además, se añaden conexiones residuales sobre las tres capas, junto con una capa de normalización. Asimismo, en la primera capa de atención se emplea enmascarado, que combinado con el hecho de que las salidas de los *embeddings* se compensan en una posición, asegura que las predicciones para la posición $i$

puedan depender únicamente de las salidas conocidas en las posiciones anteriores: $i - k$ siendo $k < i$.

Por lo tanto, el cálculo del producto punto escalado del mecanismo de autoatención [26], representado en la parte izquierda de la figura 2.3: *Scaled Dot-Product Attention*, se realiza sobre un conjunto de consultas simultáneamente encapsuladas en una matriz $Q$. Por otra parte, las llaves y los valores son encapsulados en las matrices $K$ y $V$ de dimensiones $d_k$ y $d_v$, respectivamente. Por consiguiente, la formulación de dicho cálculo es como sigue:

$$Atencin\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.11}$$

debido a que el cálculo del modelo de atención únicamente nos entrega una respuesta por cada consulta, se suele emplear el multicabezal de atención, representado en la parte derecha de la figura 2.3: *Multi-Head Attention*, que permite realizar el cálculo anterior pero a nivel de múltiples consultas y llaves simultáneas. Dicho proceso, consiste en varios cálculos atencionales en paralelo combinando el resultado para obtener un vector de contexto. El cálculo del multicabezal de atención está descrito en la siguiente fórmula:

$$Multicabezal\,(Q, K, V) = Concat\,(cabezal_1, \ldots, cabezal_h)\,W^o \tag{2.12}$$

donde

$$cabezal_i = Atencin(QW_i^Q, KW_i^K, VW_i^V) \tag{2.13}$$

donde las proyecciones son matrices de parámetros $W_i^Q \in \mathbb{R}^{d_{modelo} * d_k}$, $W_i^K \in \mathbb{R}^{d_{modelo} * d_k}$, $W_i^V \in \mathbb{R}^{d_{modelo} * v}$, $W^o \in \mathbb{R}^{d_{modelo} * hd_v}$, los cuales son aprendidos durante la etapa de entrenamiento.

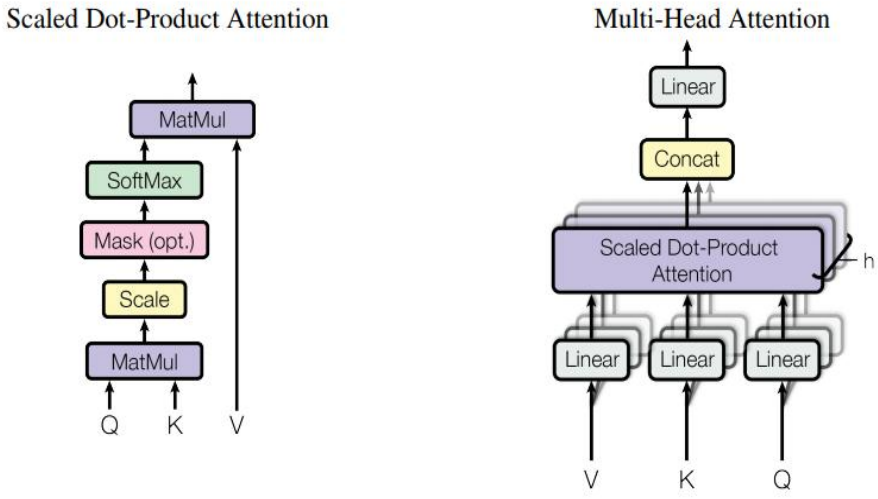


**Figura 2.3:** Mecanismo de atención con producto punto escalado (izquierda). Mecanismo multicabezal de atención con capas de atención en paralelo (derecha) [45]

.

## 2.3 Modelos de lenguaje basados en Transformer

Cuando se presentó el modelo *Transformer*, éste demostró tener una gran capacidad para codificar largas secuencias de texto. A la hora de comparar el modelo *Transformer* (ver 2.2.4) con una *RNN* (ver 2.2.1) se puede observar que el *Transformer* es totalmente,

paralelizable, no es unidirecional y no presenta liminaciones de memoria, debido a que el mecanismo de atención (ver 2.2.3) puede enfocar en el contexto relevante a la frase, permitiendo así un mayor entendimiento.

Las representaciones contextuales proporcionan unas características semánticas dinámicas de las palabras, es decir, que para una misma palabra en dos o más contextos distintos, tendrá una representación distinta.

En la actualidad, para la mayoría de tareas de NLP cuyo objetivo sea el de codificar secuencias de texto, los modelos basados en la arquitectura *Transformer* son los modelos vanguardistas. La aparición de *Bidirectional Encoder Representations from Transformers* (BERT) [10] presentó un gran impacto en el campo del NLP, debido a que proporcionó modelos de lenguaje pre-entrenados con ingentes cantidades de datos. Cabe destacar, que al realizar un refinamiento de los pesos de estos modelos para que se especialicen en una tarea concreta o en inglés *fine-tuning*, diferente a la que habían sido entrenados, pero sí que guarda cierta relación, a partir de un entrenamiento supervisado utilizando el correspondiente conjunto de datos etiquetados, hayan obtenido el mayor rendimiento hasta la fecha. Dicha metodología en la que primero se pre-entrena un modelo para posteriormente especializarlo en una tarea concreta se denomina *transfer learning* [42].

El uso de modelos de lenguaje pre-entrenados como BERT (ver 2.3.1) ha proporcionado una nueva perspectiva a la hora de resolver problemas relacionados con el campo de NLP, como puede ser la clasificación de texto[39], el análisis de sentimientos[1] o la generación de resúmenes automáticos[25]. Sin embargo, donde más ha destacado BERT ha sido en el campo de la visión por computador, en concreto, en los modelos pre-entrenados basados en redes neuronales convolucionales con el conjunto de imágenes de `ImagenNet`[1] [9].

A continuación, se comentarán las características que se propusieron en la arquitectura de *Transformer* para que fuese capaz de codificar las frases de un texto. Y, por último, se indagará en las diferencias en la fase de entrenamiento respecto a RoBERTa (ver 2.3.2), el cual es una optimización de BERT.

### 2.3.1. BERT

BERT [10] es un modelo de lenguaje pre-entrenado basado en la arquitectura de *Transformer* (ver 2.2.4) que está formado únicamente por los bloques del codificador. BERT ha sido pre-entrenado utilizando dos tareas no-supervisadas, *Masked Language Model* (MLM) y *Next Sentence Prediction* (NSP).

#### MLM

MLM es una aproximación que le permite a BERT tener bidireccionalidad, convirtiéndose así en modelo de lenguaje contextual. Para conseguir dicha bidireccionalidad, se escoge un 15 % de los tokens de una frase de forma aleatoria, los cuales se sustituyen por la etiqueta [MASK] con el objetivo de predecir dichas partes enmascaradas. Para ello, tendrá que valerse de los tokens que quedan visibles, es decir, los que no han sido enmascarados y, por tanto, el modelo estará condicionado por el contexto derecho e izquierdo para lograr el desmascaramiento de un token concreto. Por ejemplo, "la estación de [MASK] estaba llena de gente el día de antes de [MASK]". Sin embargo, ello provoca un desajuste entre el pre-entrenamiento y la especialización en una tarea concreta debido a que la etiqueta de enmascaramiento [MASK] no aparece durante el proceso de especialización. Para evitarlo, del 15 % de los tokens que van a ser enmascarados, realmente se

[1] https://image-net.org/

enmascaran únicamente el 80 %, el otro 10 % son sustituidos por una palabra aleatoria y el 10 % restante no se enmascaran.

### NSP

Además de MLM, BERT emplea en el entrenamiento otra función objetivo llamada NSP. Para cada frase de entrenamiento **x** se tiene asociada una frase **y**. El objetivo de BERT, por lo tanto, es discriminar, como si de un clasificador binario se tratase, indicando si la frase **y** es posterior a **x**, dicho de otro modo, si guardan algún tipo de relación entre sí. Resulta muy útil conocer este tipo de relaciones para algunas de las tareas de NLP, como pueden ser la inferencia del lenguaje o respuesta a preguntas. En cuanto a los datos del pre-entrenamiento, el 50 % de las veces **x** es realmente la continuación de **y** y el otro 50 % son frases asociadas aleatoriamente.

Para representar un par de frases **x** e **y** en el entrenamiento se precisa de dos etiquetas especiales. La primera es *[CLS]* que representa el principio de la frase **x** y su clasificación, y la segunda, *[SEP]*, la separación entre la frase **x** e **y**, el cual es a su vez el último token de **x**. De este modo, las dos frases de entrenamiento quedarían representadas como sigue: *[CLS] A [SEP] B*.

Al igual que en el *Transformer*, para representar la entrada de BERT se suma la representación de cada token con la correspondiente representación de la posición. Además, se suma la representación de la pertenencia del token al segmento **x** o **y**, lo cual es fundamental para la función de pérdida NSP. En la figura 2.4 se muestra dicha representación.

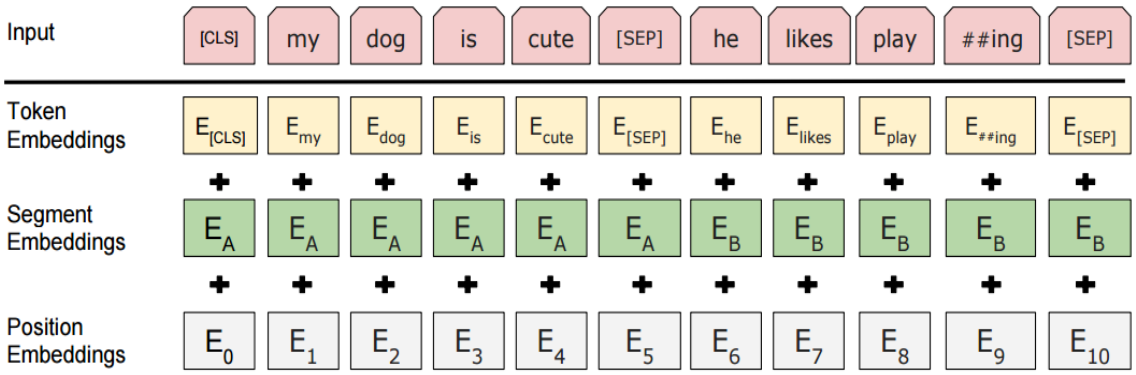


**Figura 2.4:** Representación de la entrada de BERT. Los *embeddings* de la entrada se calculan como la suma de los *embeddings* de los tokens, de lo segmentos y de las posiciones [10].

El aprendizaje de los *embeddings* se hace a partir de subpalabras generadas a la hora de segmentar las palabras en unidades más pequeñas, cuyas representaciones es lo que aprende BERT. Esto proporciona numerosas ventajas:

- Reduce de manera muy significativa el tamaño del vocabulario.
- Permite manejar mejor las palabras desconocidas.
- Las palabras que comparten una estructura parecida de subpalabras, podrían compartir una misma semántica.

El proceso de extraer subpalabras a partir de unas palabras dadas se llama tokenizar. Los tokenizadores más utilizados son el BPE (*Byte Pair Encoding*) [34] y *WordPiece* [36]. Los dos tokenizadores emplean un algoritmo iterativo donde el vocabulario es inicializado con todos los elementos que constituyen el conjunto de entrenamiento. Para el

caso del BPE, en cada iteración el vocabulario es actualizado con los pares de elementos más frecuentes. Sin embargo, en el caso del *WordPiece*, el vocabulario será actualizado por aquellos pares que obtienen mayor verosimilitud en el correspondiente conjunto de entrenamiento.

El modelo base de BERT dispone de dos configuraciones de modelos de lenguaje pre-entrenados con enormes cantidades de texto en inglés, *Base* y *Large*. Los parámetros de ambas configuraciones se pueden observar en la tabla 2.1. Como se puede observar en la tabla previamente mencionada, la configuración *Large* es la que mayor número de parámetros tiene de las dos, y por ende, con la que obtiene resultados ligeramente superiores para las distintas tareas de NLP con las que se experimentaron.

En la tabla 2.1 se muestra el valor de cada hiper parámetro de ambas configuraciones. Para cada configuración se muestra:

- Número total de parámetros del modelo.
- $L$: número de bloques de tipo codificador que lo constituyen.
- $A$: número de cabezales empleados en el multicabezal de autoatención.
- $H$: la dimensión de la representación obtenida al procesar una frase.

| **Configuración** | **Parámetros** | **L** | **H** | **A** |
|---|---|---|---|---|
| *Base* | 110M | 12 | 768 | 12 |
| *Large* | 340M | 24 | 1024 | 16 |

**Tabla 2.1:** Configuraciones del modelo base de BERT.

BERT emplea dos aproximaciones para representar las frases de entrada. La más típica consiste en utilizar la representación obtenida por el token [CLS] en el último bloque del modelo de lenguaje pre-entrenado al procesar una frase. La segunda aproximación consiste en emplear las representaciones obtenidas de este último bloque por cada uno de los tokens que forman la frase. Posteriormente, calcula el valor medio o valor máximo de cada uno de los $H$ componentes de todas las representaciones.

La principal limitación que tiene BERT es que no puede procesar secuencias de una longitud mayor a 512 tokens. En dicho caso, se tiene que truncar la secuencia. Para ello, BERT tiene tres maneras de hacer el truncamiento. La primera sería quedándose únicamente con los primeros 512 tokens. La segunda guarda relación con la primera, puesto que en este caso BERT se queda únicamente con los 512 tokens del final de la secuencia. Y la última aproximación, consiste en tomar solamente los tokens del principio y del final, sin tener en cuenta los tokens del medio.

### 2.3.2. RoBERTa

El modelo RoBERTa (*Robustly optimized BERT approach*) [51] es una extensión de BERT. Dicha extensión aplica algunas modificaciones en ciertos aspectos del entrenamiento con el objetivo de mejorar el rendimiento obtenido por BERT en algunas de las tareas de NLP.

Para poder cumplir el objetivo anterior, se elimina la función objetivo NSP, puesto que se ha demostrado, gracias a diferentes experimentos, que no es óptima. En consecuencia, los tokens especiales [CLS] y [SEP] se eliminan y se añaden otros dos tokens especiales. Por una parte, el primer token de **x** es el token especial <s> y, a su vez, el último token de **x** es el token especial </s>, de este modo, quedaría como: <s>$x$</s>. Por otra parte,

la función objetivo MLM se modifica ligeramente, debido a que en RoBERTa se aplica enmascaramiento dinámico, en vez de estático, como ocurre en BERT. Ello conlleva que se apliquen 10 enmascaramientos distintos, causando que el corpus de entrenamiento sea 10 veces más amplio que el inicial. Esto muestra, en los diferentes experimentos realizados, que esta aproximación dinámica del MLM obtiene mejoras en el rendimiento con respecto de la aproximación estática de BERT.

Además, los hiperparámetros empleados durante el entrenamiento de RoBERTa emplean tamaños de *minibatch* y de factor de aprendizaje cuyo valor es de mayor valor que los utilizados en BERT. La longitud máxima de la secuencia de entrada del modelo de entrenamiento también se ve aumentada.

Por último, comentar un par de modificaciones más realizadas en el modelo de RoBERTa. El tamaño del vocabulario se ve aumentado de 30k subpalabras a 50k subpalabras. Además, se emplea un conjunto de entrenamiento inmensamente mayor respecto al empleado para entrenar el modelo de BERT. RoBERTa emplea, además del conjunto de datos que emplea BERT *BookCorpus* [50] de 16GB, tres conjuntos de datos adicionales: *CC-NEWS* [2] de 76GB, *OpenWebText* [3] de 38GB y *Stories* [44] de 31GB. En total, el conjunto de datos de RoBERTa es de 161GB, frente al de BERT que es únicamente de 16GB, lo que supone una diferencia abismal de datos.

Con todas las modificaciones comentadas, RoBERTa consigue una mejora del 15 % respecto a BERT en el *SuperGLUE benchmark* [46]. Dicho *benchmark* consiste en varias tareas de NLP como por ejemplo, clasificación de texto, contestación a preguntas o análisis de sentimientos. Debido a que se va a emplear un modelo de lenguaje pre-entrenado para la generación de datos a partir de los ya existentes, y RoBERTa da mejor rendimiento del modelo con las mismas dimensiones, se va a emplear RoBERTa y, para ser más exactos, un modelo de lenguaje pre-entrenado de RoBERTa *fine-tuned* al ruso (ruRoberta-large[4]). El coste de aplicar dicho *fine-tuning* a un modelo de lenguaje pre-entrenado de BERT o RoBERTa es prácticamente el mismo. Sin embargo, hay que remarcar que para RoBERTa el hecho de incorporar modificaciones en el modelo de lenguaje pre-entrenado tiene un mayor coste de pre-entrenamiento respecto a BERT.

### 2.3.3. GPT-2

GPT-2 (*Generative Pre-trained Transformer 2*) es la segunda versión de la serie de modelos GPT basados en la arquitectura *Transformer* la cuál aumenta la cantidad de hiperparámetros de entrenamiento que se emplean hasta 1.5B frente a los 117M hiper parámetros del modelo previo, GPT-1 [31]. La relación entre la cantidad de hiper parámetros empleados y de la dimensionalidad de los distintos tipos de modelos de GPT-2 se refleja en la tabla 2.2. Cabe destacar que, GPT está formado de múltiples capas de decodificador a diferencia de BERT (ver 2.3.1) que está formado de capas de codificador.

El conjunto de datos empleados en el entrenamiento de GPT-2, está formado por textos extraídos de páginas *web*, centrados en la calidad de los mismos, de más de cuarenta y cinco millones de direcciones de enlaces, el conjunto de datos fue bautizado como *WebText* [32]. Aunque dicho conjunto de datos no ha sido lanzado públicamente todavía, se puede afirmar sin ninguna duda, que la selección de los textos se ha visto comprometida a ciertos criterios de calidad y que está formado, por una gran variabilidad de dominios. Esto se puede ver reflejado en la capacidad del modelo de conseguir resultados de estado de arte en diferentes tareas de NLP sin necesidad de entrenamiento adicional. Por

[2] https://commoncrawl.org/2016/10/news-dataset-available/
[3] https://skylion007.github.io/OpenWebTextCorpus/
[4] https://huggingface.co/sberbank-ai/ruRoberta-large

| Parámetros | Capas | Dimensionalidad |
|---|---|---|
| 117M | 12 | 768 |
| 345M | 24 | 1024 |
| 762M | 36 | 1280 |
| 1542M | 48 | 1600 |

**Tabla 2.2:** Hiper parámetros de arquitectura para las cuatro dimensionalidades del modelo GPT [32].

ejemplo, en la tabla 2.3 se puede observar como en la tarea de traducción automática del francés al inglés se consiguen resultados de vanguardia si se da como entrada una frase apropiada.

| |
|---|
| "I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool].** |
| In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: **"Mentez mentez, il en restera toujours quelque chose,"** which translates as, **"Lie lie and something will always remain."** |
| "I hate the word **'perfume,'**" Burr says. 'It's somewhat better in French: **'parfum'**. |
| If listened carefully at 29:55, a conversation can be heard between two guys in French: **"-Comment on fait pour aller de l'autre cote? -Quel autre coté?"**, which means **"- How do you get to the other side? - What side?"**. |
| If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater? |
| **"Brevet Sans Garantie Du Gouvernement"**, translated to English: **"Patented without government warranty"**. |

**Tabla 2.3:** Ejemplos de traducción empleando GPT-2 sin entrenamiento adicional [32].

El modelo más grande de GPT-2, el *GPT-2 extra large* tiene una ventana de contexto $n$ de dimensión 1600 la cual se comparte con la salida conforme se va generando el texto. En cuanto al modelo original de GPT-2, éste fue entrenado con un corpus de 40GB llamado WebText.

Por último, cabe destacar que la serie de modelos GPT anunció recientemente el mayor modelo que han entrenado hasta la fecha, el GPT-3 [15] que actualmente es el modelo más potente de la serie de modelos GPT.

### 2.3.4. XLNet

XLNet [49] es una aproximación de modelo de lenguaje pre-entrenado basado en *Transformer* que incorpora regresión automatica como GPT-2 (ver 2.3.3) y encuentra una alternativa para incorporar contexto en ambos sentidos como BERT (ver 2.3.1).

Por lo tanto, el objetivo que persigue es el llamado modelado del lenguaje de permutación, el cual además de conservar los beneficios de la autoregresión, también permite que los modelos capturen contexto bilateral. Así, para una secuencia $\mathbf{x}$ de longitud $T$, existen $T!$ diferentes órdenes para realizar un factorización autorregresiva. Se puede deducir que, si los parámetros del modelo están compartidos en todos los órdenes de factorización, se puede esperar que el modelo aprenderá a recopilar la información de todas las posiciones en ambos lados [49].

## 2.4 Técnicas de Aumento de Datos

En esta última sección del apartado del estado del arte, se van a exponer algunas de las técnicas existentes hoy en día para la generación de nuevos textos sintéticos a partir de los textos de entrada. Además, se va a justificar la necesidad y exponer las ventajas de emplear dichas técnicas para generar datos para la tarea de traducción automática. En los siguientes subapartados se van a explicar tres técnicas que han sido seleccionados por su popularidad en el campo de NLP para el aumento de los datos. La primera de ellas será *back-translation* (ver 2.4.1), la segunda será el aumento contextual que realiza sustituciones de palabras utilizando modelos de lenguaje pre-entrenados (ver 2.4.2) y la última será *Easy Data Augmentation* (ver 2.4.3).

### 2.4.1. Back-translation

La primera técnica para la generación de datos que se va a comentar es el *back-translation* [14]. Para emplear dicha técnica, se precisa de dos idiomas, el idioma relativo a las frases originales $X$ y el idioma secundario $Y$.

El proceso que se sigue para llevar a cabo este método es el siguiente. Dada una frase en el idioma $X$ y empleando un modelo de traducción automática, se traduce del idioma del texto $X$, obteniendo un texto del idioma del texto $Y$. A continuación, se emplea la nueva frase generada $Y$ para hacer otra traducción, pero esta vez al idioma del texto original $X$. El objetivo de este procedimiento es obtener una frase en la cual *a priori* se mantiene el significado semántico de la frase original pero con una estructura y/o palabras distintas, se puede decir que se trata del parafraseo del texto original $X$. En algunos casos, la frase generada es idéntica a la original, el hecho de que esto suceda depende de la complejidad de la frase a traducir y del rendimiento de los traductores empleados, en dicho caso simplemente se descartaría la frase generada puesto que no aportaría nada al entrenamiento.

Para exponer el uso de este método para aumento de datos se va a utilizar el ejemplo empleado en el artículo [48] donde se emplean dos modelos de traducción automática para el par de idiomas Inglés-Francés, los cuales han sido entrenados con el conjunto de datos de la conferencia *WMT'14* [5] (*Workshop on Statistical Machine Translation*). Además, conservan los traductores y no únicamente emplean el mejor obtenido en el entrenamiento. Con el objetivo de generar traducciones un tanto distintas, emplean dichos traductores para cada sentido y variando el valor de la temperatura del decodificador lo cual permite

[5] https://dl.acm.org/conference/wmt

alterar el espacio de búsqueda en la generación de texto. En la tabla 2.4 se muestran los diversos ejemplos que han logrado obtener, los cuales además conservan el significado semántico original, por ello la calidad de los textos generados al emplear este método es alta.

| | |
|---|---|
| **Original** | Given the low budget and production limitations, this movie is very good. |
| ***Back-translation*** | Since it was highly limited in terms of budget, and the production restrictions, the film was cheerful.<br>There are few budget items and production limitations to make this film a really good one.<br>Due to the small dollar amount and production limitations the ouest film is very beautiful. |

**Tabla 2.4:** Ejemplos de textos generados obtenidos al aplicar la técnica de *back-translation* en traductores automáticos Inglés-Francés.

El mayor inconveniente de emplear esta técnica es que se precisa, como mínimo, de dos modelos de traducción automática. Además, para poder emplear dichos modelos se precisa de un equipo con la potencia suficiente y de un *software* concreto que haga posible la obtención de las traducciones en un tiempo razonable. Por ello, la manera más sencilla y rápida es emplear algún servicio en la nube que mediante peticiones, por ejemplo a través de una API, devuelva las traducciones correspondientes, aunque en la mayoría de estos servicios existe un límite de peticiones por unidad de tiempo, lo que supondría tener algún tipo de plan de pago para poder explotar al máximo dichos servicios. Dependiendo de las necesidades del usuario, se pueden emplear los servicios gratuitos pero limitados o bien los servicios de pago pero con mayor cantidad de peticiones. Sin embargo, otra alternativa sería emplear un modelo de traducción automático entrenado, ya sea empleando recursos propios o empleando servicios en la nube que permiten realizar ejecuciones que requieren un alto rendimiento computacional y espacial.

### 2.4.2. Sustitución de palabras empleando modelos de lenguaje pre-entrenados

La segunda aproximación es la técnica principal empleada en este proyecto para el generación de textos en ruso. Dicha aproximación consiste en el uso de un modelo de lenguaje pre-entrenado para generar datos nuevos a partir de los ya existentes. En concreto, se ha empleado para el presente proyecto un modelo basado en RoBERTa (ver 2.3.2).

El método de aumento de datos empleado en este proyecto, está basado en la primera tarea con la que se entrena BERT, comentada en este escrito (ver 2.3.1), la cual consiste en enmascarar una palabra de la frase, dejando el resto de palabras intactas, y pasarle como parámetro de entrada la frase con la palabra enmascarada a BERT, el cual actuará como modelo de lenguaje contextual dando como resultado $C$ candidatos (normalmente se usan C = 15, pero en este proyecto se ha limitado a 5 el número de candidatos debido a que la sustitución se hace con un *POS-tagger*, donde *POS* viene del inglés *Part Of Speech* y hace referencia a un tipo de clasificación que puede definirse como la asignación automática de una etiqueta, que puede representar una de las partes del habla, información semántica o otro tipo de información relevante que pueda tener el token en cuestión, lo cual disminuye el abanico de posibilidades para BERT de encontrar una palabra válida). Además, cada candidato tiene una probabilidad, dada por BERT, de pertenecer al contexto de la frase, $p_m$ ($p_m = 0{,}4$). La selección de la palabra enmascarada viene dada por el tipo de aumento de dato que se quiera realizar (en base al tipo de palabra: sustantivo, adverbio, pronombre o una mezcla de las anteriores). Una vez seleccionado el tipo de

aumento de datos que se quiera emplear con el modelo de lenguaje, se elige de forma aleatoria dentro de la lista de las palabras de dicho tipo extraídas de la frase. Si se desea, se pueden cambiar más de una o incluso todas las palabra de la lista, debido a que se almacena la posición de la palabra en la frase, para evitar así sustituciones múltiples sobre el mismo índice en la frase.

Sin embargo, existen otras técnica de aumento de datos basadas en la tarea de *MLM* como podría ser la empleada en el artículo de TinyBERT [19]. Dicho método consiste en sustituir una palabra de la frase por la etiqueta de enmascarado para seguidamente utilizar BERT como un modelo de lenguaje para obtener los $C$ candidatos (usualmente $C = 15$) más probables de ocupar esa posición en la frase, sin modificar en absoluto el resto de palabras de la frase. Además, la palabra enmascarada tiene una probabilidad $p_m$ (usualmente $p_m = 0{,}4$) de ser sustituida por una de las candidatas seleccionadas de forma aleatoria, es decir, si la probabilidad de la palabra aleatoria $p_a$ es inferior a $p_m$, la palabra aleatoria no se selecciona para ser la sustituta de la palabra enmascarada. Para generar la nueva frase, se ha de iterar sobre todas las palabras de la frase, por lo que, por cada pasada de esta técnica, únicamente se obtendrá un ejemplo aumentado. Por ello, se realizan $N$ pasadas para así poder obtener $N$ ejemplos nuevos.

### 2.4.3. Easy Data Augmentation

La última técnica de aumento de datos que se va a comentar es *Easy Data Augmentation* (EDA) [47]. Debido a que esta técnica realiza cambios muy simples y que no requieren de mucho tiempo ni capacidad de cómputo para generar una frase sintética nueva, supone un coste muy reducido y, además, es muy rápida. Este método propone cuatro funciones de transformación para generar texto sintético, las cuales se comentan a continuación:

- *Synonym Replacement (SR)*. Se seleccionan aleatoriamente $N$ palabras del texto original, filtrando las palabras que sean *stopwords*. Cada una de las palabras seleccionadas se sustituyen aleatoriamente por uno de sus correspondientes sinónimos.
- *Random Insertion (RI)*. Se selecciona aleatoriamente un sinónimo de la frase filtrando las palabras que sean *stopwords*. Posteriormente, se inserta dicho sinónimo en una posición aleatoria de la frase. Se realiza este procedimiento $N$ veces.
- *Random Swap (RS)*. Se selecciona aleatoriamente dos palabras de la frase y se intercambian de posición entre ellas. Se realiza este procedimiento $N$ veces.
- *Random Deletion (RD)*. Se eliminan palabras de la frase que tengan una probabilidad igual a $p$.

La relación de inducción de ruido se elige en función de la longitud de la frase. La hipótesis que es propuesta por los autores es que los documentos cortos son menos propensos a la manipulación debido al ruido y la transformación puede resultar en una variabilidad de clase. Debido a esto, las frases largas pueden absorber más cantidad de ruido mientras mantienen la etiqueta de clase original.

Dicha relación de inducción se aplica a las técnicas descritas anteriormente, salvo la primera de todas ($SR$). Esta técnica requiere de dos hiper parámetros. El primero de ellos es la longitud de la frase a nivel de palabra, $L$ y el segundo es $\alpha$, el cual indica el porcentaje de palabras que se van a cambiar en una frase. Lo que se hace es jugar con el valor de $\alpha$, los valores que se suelen emplear para la experimentación son $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Con esto se consigue la siguiente fórmula $n = \alpha L$ la cual define la cantidad de palabras que

se van a cambiar en la frase. En la figura número tres del artículo [47], se puede observar la evolución de las distintas técnicas de EDA variando el parámetro $\alpha$ y la cantidad del conjunto de datos de entrenamiento empleado. En dicha figura, puede apreciarse que las cuatro técnicas de aumento de datos empleadas contribuyen al aumento de las ganancias de rendimiento. Así:

- Empleando la técnica *SR*, la mayor mejora de rendimiento se ve con valores pequeños de $\alpha$, pero con valores grandes, el rendimiento se ve mermado. Debido a que al reemplazar demasiadas palabras en la frase, la identidad del texto se ve cambiada.
- Empleando la técnica *RI*, las ganancias de rendimiento fueron más estables para diferentes valores de $\alpha$, seguramente porque las palabras originales en la frase y su orden relativo se mantuvieron.
- Empleando la técnica *RS*, produjo ganancias de rendimiento en $\alpha \leq 0.2$, pero disminuyó en $\alpha \geq 0.3$ ya que realizar demasiados intercambios equivale a barajar demasiado el orden de la frase.
- Empleando la técnica *RD*, se tenía las ganancias más altas para una $\alpha$ baja pero perjudican gravemente el rendimiento a altas $\alpha$, ya que las frases son probablemente ininteligibles si se eliminan hasta la mitad de las palabras.

En la tabla 2.5 se pueden observar ejemplos concretos de las cuatro técnicas de EDA descritas en este apartado.

| **Operation** | **Sentence** |
|---|---|
| None | A sad, superior human comedy played out on the back roads of life. |
| SR | A ***lamentable***, superior human comedy played out on the ***backward*** road of life |
| RI | A sad, superior human comedy played out on ***funniness*** the back roads of life. |
| RS | A sad, superior human comedy played out on ***roads*** back ***the*** of life. |
| RD | A sad, superior human out on the roads of life. |

**Tabla 2.5:** Ejemplos de textos generados obtenidos al aplicar las distintas técnicas de EDA [47].

# CAPÍTULO 3
# Generación automática de corpus inglés-ruso

En el siguiente capítulo, se van a comentar los modelos de lenguaje pre-entrenados empleados en el presente proyecto. A continuación, se van a comentar las técnicas empleadas para la generación de nuevos conjuntos de datos paralelos de inglés-ruso, así como ejemplos específicos de cada técnica y las estadísticas que se han extraído.

## 3.1 modelos de lenguaje pre-entrenados para ruso

Como se ha comentado en la sección 2.3.2, RoBERTa es una de las mejores opciones para la tarea de generación de texto contextual. Por lo tanto, tras realizar experimentos con distintos modelos de lenguaje pre-entrenados, entre ellos BERT 2.3.1 y GPT-2 2.3.3, se ha escogido, para el desarrollo del presente proyecto, un modelo de lenguaje pre-entrenado basado en RoBERTa con *fine-tuning* para el idioma ruso, llamado *ruRoberta-large*[1]. Se ha empleado la técnica de *fill mask*, comentada en la sección 2.3.1 que consiste en enmascarar un token de la frase y utilizar el modelo de roBERTa para que intente averiguar, según el contexto y empleando *word embeddings*, la palabra en la posición de la etiqueta de enmascaramiento, cuidando de que la palabra generada por el modelo no sea la misma que la que había antes. Además, se respeta que el *tag* de la palabra sea el mismo, es decir, que la clase de la palabra original y la generada tiene que coincidir, esto se hace mediante un *POS-tagger* especializado en ruso desarrollado por Spacy[2] llamado *ru_core_news_lg*[3] y la librería *pymorphy2*[4].

Además, el modelo de lenguaje pre-entrenado empleado en la generación de datos devuelve un valor por cada token generado, el cual indica la pertenencia de dicho token al contexto dado de la frase. A raíz de este valor, se han realizado sucesivos experimentos para determinar empíricamente el valor del umbral de aceptación de un token y se ha llegado a la conclusión de establecer dicho umbral al valor 0.01. El texto en ruso en alfabeto cirílico se ha tratado en UTF-8 sin emplear métodos de transliteración.

[1] https://huggingface.co/sberbank-ai/ruRoberta-large
[2] https://spacy.io/
[3] https://spacy.io/models/ru#ru_core_news_lg
[4] https://pymorphy2.readthedocs.io/en/stable/user/index.html

## 3.2 Técnicas de Aumento de Datos

Hay que tener en cuenta que el cambio de uno o varios tokens en una frase puede llegar a suponer una alteración en el significado inicial de la frase. Sin embargo, debido a que nuestra intención es crear datos específicos para entrenar modelos de traducción automática de carácter general, la variación de dominio al generar nuevas frases no supone un problema. Aunque, si que es de suma importancia, que la frase final, es decir, la frase resultante de las $n$ sustituciones sucesivas de tokens que se hagan en la frase inicial, tiene que tener cierto grado de coherencia. Para lograr esto, las sustituciones que se realizan para generar la nueva frase, se realizan de manera secuencial, es decir, partiendo de la frase inicial, se identifica la posición de cada uno de los tokens, con el objetivo de identificar la cantidad de posibles candidatos que se van a cambiar y sus posiciones en la frase. A continuación, se selecciona un candidato a ser sustituido, comprobando que dicho candidato o, más bien, su posición no se hayan seleccionado en alguna anterior iteración, y se le asigna la etiqueta de enmascaramiento, $< mask >$, para que el modelo de lenguaje pre-entrenado sepa donde posicionar la nueva palabra generada, seleccionada a partir de los criterios específicos de cada tipo de técnica de aumento de datos empleadas en el presente proyecto (ver 3.2.1, 3.2.2, 3.2.3), en la nueva frase, la cual se volverá a pasar por el mismo proceso hasta que se terminen los candidatos. Con ello, conseguimos que la frase final generada cumpla con el criterio de coherencia al no permitir sustituciones en paralelo, que corromperían el análisis del contexto de la frase, realizado por el modelo de lenguaje pre-entrenado, al seleccionar una palabra.

### 3.2.1. Sustitución de adverbios

La primera aproximación que se propone en el presente trabajo es la sustitución de varios adverbios de la frase original por otros empleando el modelo de lenguaje pre-entrenado *ruRoBERTa-large*. La motivación que sigue a esta técnica es la gran cantidad de variabilidad y riqueza lingüística que proporcionan los adverbios presentes en el ruso, lo cual facilita la creación de nuevos textos.

Por un lado, el objetivo último de esta aproximación es cambiar la máxima cantidad de adverbios en la frase original, con la meta de crear mayor variabilidad sin alterar significativamente los detalles contextuales y manteniendo inalterables los detalles gramaticales de la frase original. A la hora de elegir el candidato que sustituya a la palabra original, se emplea el *POS-tagger* de Spacy comentado en el apartado 3.1 para determinar que la clase de la palabra en la palabra original y la generada sea un adverbio. Además, en la Tabla 3.1 se pueden encontrar ejemplos específicos de generación de datos empleando dicha técnica.

Por último, en la tabla 3.2 se muestran las estadísticas que se han recogido a partir de la generación de nuevas frase empleando la técnica de sustitución de adverbios. En dicha tabla, se puede observar que se han generado más de la cuarta parte de la cantidad inicial de datos empleados. Además, en este caso la tasa de palabras generadas es baja, teniendo aproximadamente una palabra nueva por frase generada.

| Original | | Generado | |
|---|---|---|---|
| **Inglés** | **Ruso** | **Inglés** | **Ruso** |
| **Further**, violence against women, which **often** accompanies forced evictions, has been recognized by the Committee on the Elimination of Discrimination against Women to be a form of discrimination against women. | Далее, Комитет по ликвидации дискриминации в отношении женщин признал, что насилие в отношении женщин, которым часто сопровождаются принудительные выселения, является одной из форм дискриминации в отношении женщин. | **Finally**, the Committee on the Elimination of Discrimination against Women acknowledged that violence against women who are **usually** accompanied by forced evictions is one form of discrimination against women. | Наконец, Комитет по ликвидации дискриминации в отношении женщин признал, что насилие в отношении женщин, которым обычно сопровождаются принудительные выселения, является одной из форм дискриминации в отношении женщин. |
| Their **greatest** concentration is in the Kingston Metropolitan Area (KMA). | Наиболее высокая концентрация такого жилья отмечалась в столичном округе Кингстон. расхождениях не поступало. | A **particularly** high concentration of such housing was noted in the capital district of Kingston. | Особенно высокая концентрация такого жилья отмечалась в столичном округе Кингстон. |
| The Group of 77 and China **firmly** believed that Agenda 21 should continue to be **implemented** at the same time as the recommendations of the major United Nations conferences held since 1992, **especially** the Johannesburg Plan of Implementation. | Группа 77 и Китай твердо убеждены в том, что практическую реализацию Повестки дня на XXI век следует продолжать наряду с осуществлением рекомендаций крупных конференций Организации Объединенных Наций, которые были проведены с 1992 года, особенно Плана выполнения решений Йоханнесбургской встречи. | The Group of 77 and China are **also** convinced that the practical implementation of the seabed Agenda for the XXI century should continue in **parallel** with the implementation of the recommendations of major United Nations conferences that have been held since 1992, **including** the Plan of Implementation of the Johannesburg Meeting. | Группа 77 и Китай также убеждены в том, что практическую реализацию Повестки дня на XXI век следует продолжать параллельно с осуществлением рекомендаций крупных конференций Организации Объединенных Наций, которые были проведены с 1992 года, включая Плана выполнения решений Йоханнесбургской встречи. |

**Tabla 3.1:** Ejemplos de textos generados obtenidos al sustituir adverbios.

| Estadísticas | Valor |
|---|---|
| Cantidad de frases originales | 1M |
| Cantidad de frases nuevas | 275K |
| Porcentaje de frases nuevas | 27,5 % |
| Cantidad total de palabras nuevas | 327K |
| Media de palabras generadas por frase | 1,119 |
| Tiempo empleado | 5 días |

**Tabla 3.2:** Estadísticas recogidas de la generación de nuevos datos por sustitución de adverbios empleando el modelo de lenguaje pre-entrenado *ruRoBERTa-large*. La M hace referencia a los millones y la K a los miles.

### 3.2.2. Sustitución de sustantivos

La segunda aproximación que se propone en el presente trabajo es la sustitución de sustantivos de la frase original por otros empleando el modelos de lenguaje pre-entrenado *ruRoBERTa-large*. Aprender los sustantivos en ruso es significativamente importante, porque su estructura se utiliza en todas las conversaciones diarias. Es de vital importancia que en la frase haya sustantivos apropiados para que la frase tenga fluidez. Por tanto, la motivación que sigue a esta técnica es proporcionar, a partir de las frases originales, nuevas frases manteniendo la fluidez de la frase original.

Por un lado, a parte de emplear el *POS-tagger* de *Spacy* comentado en el apartado anterior, se emplea además la librería de *pymorphy2* para poder conocer de forma pragmática el género y número de la palabra que se va a cambiar, para que, al seleccionar la palabra que la va a sustituir, se cumpla que tengan el mismo número y género. Por ello, se puede mantener la integridad de la frase, que es el objetivo principal de esta técnica.

En la tabla 3.3 se pueden observar ejemplos de frases creadas a partir de una original empleando la técnica de sustitución de sinónimos.

Por último, en la tabla 3.4 se muestran las estadísticas que se han recogido a partir de la generación de nuevas frases empleando la técnica de sustitución de sustantivos. En dicha tabla, se puede observar que se ha generado casi la misma cantidad de datos, empleando el conjunto de datos inicial. Además, se puede observar que la media de palabras generadas es de aproximadamente cinco palabras por frase nueva.

| Original | | Generado | |
|---|---|---|---|
| **Inglés** | **Ruso** | **Inglés** | **Ruso** |
| **The purpose** of this **provision** is to **ensure** the effective restoration of the status of habitual **residents** as protected under paragraph 1. | Цель этого положения заключается в обеспечении действительного восстановления статуса обычно проживающих лиц, защищаемого согласно пункту 1. | **Part of** this **law** is the **need** for valid **compliance** with the status of usually resident **citizens**, protected under paragraph 1. | Часть этого закона заключается в необходимости действительного соблюдения статуса обычно проживающих граждан, защищаемого согласно пункту 1. |
| No major **discrepancies** have been **reported**, however. | Однако никаких сообщений о крупных расхождениях не поступало. | However, no **data** on the large **victims** were received. | Однако никаких данных о крупных пострадавших не поступало. |
| Many **peasants** use their agricultural products mainly for their own **needs**; only a small **share** is produced for the **market**, on **average** from 30 to 40 per cent. | Многие крестьяне используют свою сельскохозяйственную продукцию главным образом для удовлетворения своих собственных потребностей; лишь небольшая ее часть, в среднем 30-40%, производится для сбыта. | Many **people** use their agricultural products mainly to **ensure** their own **needs**; only a small **proportion** of it, in the **region** of 30-40 %, is produced for **export**. | Многие народы используют свою сельскохозяйственную продукцию главным образом для обеспечения своих собственных нужд; лишь небольшая ее доля, в районе 30-40%, производится для экспорта. |

**Tabla 3.3:** Ejemplos de textos generados obtenidos al sustituir sustantivos.

| Estadística | Valor |
|---|---|
| Cantidad de frases originales | 1M |
| Cantidad de palabras nuevas | 4,771M |
| Cantidad de frases nuevas | 922K |
| Porcentaje de frases nuevas | 92,16 % |
| Media de palabras generadas por frase | 5,18 |
| Tiempo empleado | 8 días |

**Tabla 3.4:** Estadísticas recogidas de la generación de nuevos datos por sustitución de sustantivos empleando el modelo de lenguaje pre-entrenado *ruRoBERTa-large*. La M hace referencia a los millones y la K a los miles.

### 3.2.3. Sustitución de adjetivos

La siguiente técnica de aumento de datos que se propone en el presente trabajo es la sustitución de adjetivos en la frase original. Los adjetivos en el ruso, y como en prácticamente todos los idiomas, son una parte fundamental de la riqueza de una frase. La elección correcta de un adjetivo, o un conjunto de los mismos, para describir una acción, lugar, persona, etc. es una tarea muy importante. Por ello, la motivación de está técnica es de seleccionar los adjetivos apropiados que se van a sustituir por los seleccionados de la frase original.

A la hora de elegir el candidato que sustituya al adjetivo original, al igual que en el caso de los sustantivos, se tiene en cuenta que tanto el género como el número coincidan. Además, con la ayuda del *POS-tagger* de *Spacy* especializado en ruso, comentado en los apartados anteriores, se asegura que la palabra original y generada sean ambas un sustantivo.

En la tabla 3.5 se pueden observar ejemplos de frases creadas a partir de una original empleando la técnica de sustitución de sinónimos.

Por último, en la tabla 3.6 se muestran las estadísticas que se han recogido a partir de la generación de nuevas frase empleando la técnica de sustitución de adjetivos. En dicha tabla, se puede observar que se han generado alrededor del 80 % de la cantidad inicial de datos empleados. Además, en este caso la tasa de palabras generadas es relativamente alto, teniendo casi tres palabras por frase generada.

| Original | | Generado | |
|---|---|---|---|
| **Inglés** | **Ruso** | **Inglés** | **Ruso** |
| The conclusion of the Treaty of **Pelindaba** denotes the **firm** will of the **African States** to rid the continent of the spectre of nuclear weapons, and their **steadfast** commitment to contribute to achieving the goals of **worldwide** nuclear non-proliferation and disarmament. | Заключение Пелиндабского договора свидетельствует о твердой решимости африканских государств избавить континент от призрака ядерного оружия, а также о неизменной приверженности содействию достижению целей глобального ядерного нераспространения и разоружения. | The conclusion of this **treaty** indicates the **full** determination of **European states** to rid the continent of the ghost of **chemical** weapons, as well as a **strong** commitment to promoting the achievement of the goals of **international** nuclear non-proliferation and disarmament. | Заключение Данного договора свидетельствует о полной решимости европейских государств избавить континент от призрака химического оружия, а также о твердой приверженности содействию достижению целей международного ядерного нераспространения и разоружения. |
| The working conditions in the **tobacco** and **wine-making** industries, where the air is polluted with **harmful** gases and steam, leave much to be desired. | Условия работы на предприятиях табачной и винодельческой промышленности, в которых воздух загрязнен вредными газами и паром, оставляют желать много лучшего. | Working conditions at **chemical** and **oil** industry enterprises, in which air is contaminated with **various** gases and steam, leave much better to be desired. | Условия работы на предприятиях химической и нефтяной промышленности, в которых воздух загрязнен различными газами и паром, оставляют желать много лучшего. |
| In implementing the decisions of the Governing Council, the Programme will help reduce duplication in the United Nations system by forging stronger **partnerships** with **other concerned** agencies and programmes. | При выполнении решений Совета управляющих Программа будет способствовать сокращению масштабов дублирования в системе Организации Объединенных Наций путем укрепления партнерских связей с другими заинтересованными учреждениями и программами. | In implementing the decisions of the Governing Council, the Program will contribute to reducing duplication in the United Nations system by strengthening **direct** links with **various government** agencies and programs. | При выполнении решений Совета управляющих Программа будет способствовать сокращению масштабов дублирования в системе Организации Объединенных Наций путем укрепления прямых связей с различными государственными учреждениями и программами. |

**Tabla 3.5:** Ejemplos de textos generados obtenidos al sustituir adjetivos.

| Estadística | Valor |
|---|---|
| Cantidad de frases originales | 1M |
| Cantidad de palabras nuevas | 2,144M |
| Cantidad de frases nuevas | 800K |
| Porcentaje de frases nuevas | 80 % |
| Media de palabras generadas por frase | 2,68 |
| Tiempo empleado | 6 días |

**Tabla 3.6:** Estadísticas recogidas de la generación de nuevos datos por sustitución de adjetivos empleando el modelo de lenguaje pre-entrenado *ruRoBERTa-large*. La M hace referencia a los millones y la K a los miles.

### 3.2.4. Sustitución mixta

En esta última técnica que se propone en el presente trabajo, se propone elaborar un conjunto de datos empleando las tres técnicas de aumento de datos descritas en este capítulo. Por lo tanto, con el objetivo de presentar un conjunto de datos lo más variado posible en el proyecto, se ha generado un conjunto de datos paralelos utilizando las tres técnicas de aumento de datos mencionadas en los apartados anteriores. La estrategia que se ha seguido es la siguiente: para cada frase, se selecciona entre los métodos de aumento de datos empleados, siguiendo una distribución equitativa discreta donde en por cada frase se selecciona una técnica de aumento de datos. Además, con el objetivo de que haya un porcentaje equilibrado de técnicas empleadas para la generación del conjunto de datos, dicho método no va a poder ser seleccionado en las próximas frases, hasta que no se utilicen las técnicas restantes.

En la tabla 3.7 se pueden observar ejemplos de frases creadas a partir de una original empleando la técnica de sustitución mixta.

Por último, en la tabla 3.8 se muestran las estadísticas que se han recogido a partir de la generación de nuevas frase empleando la técnica de sustitución mixta. En dicha tabla, se puede observar que se han generado alrededor del 80 % de la cantidad inicial de datos empleados. Además, en este caso la tasa de palabras generadas es alta, teniendo aproximadamente cuatro palabras nuevas por frase generada.

| Original | | Generado | |
|---|---|---|---|
| **Inglés** | **Ruso** | **Inglés** | **Ruso** |
| This approach has worked **successfully** in even the most difficult of circumstances. | Этот подход к исполнению программ срабатывал весьма успешно даже в самых сложных обстоятельствах. | This approach to program execution worked **very effectively** even in the most difficult circumstances. | Этот подход к исполнению программ срабатывал очень эффективно даже в самых сложных обстоятельствах. |
| As provided for in the **report** of the Executive Director contained in **document** DP/1994/62, **staff** holding UNDP appointments that are not limited to UNOPS will continue to hold the same letters of appointment. This approach has worked **successfully** in even the most difficult of circumstances. | Как предусматривается в докладе Директора-исполнителя, содержащемся в документе DP/1994/62, сотрудники, имеющие контракты ПРООН, не ограничиваемые назначением в ЮНОПС, сохранят за собой те же контракты. | As envisaged in the **status** of the Presidential Director, contained in the **law** DP/1994/62, **people** with UNDP contracts not limited to work in UNOPS will retain the same contracts. | Как предусматривается в статусе Директора-президента, содержащемся в законе DP/1994/62, люди, имеющие контракты ПРООН, не ограничиваемые работой в ЮНОПС, сохранят за собой те же контракты. |
| The panellists will be Mr. Shabbir G. Cheema, Director, **Management** Development and **Governance** Division, UNDP, and Ms. Cheryl Gray, Manager of **Public Sector** Management Unit, World Bank. | Чима, Директор Отдела по развитию управленческих кадров и административным вопросам ПРООН, и г-жа Шерил Грей, руководитель Группы по вопросам управления в государственном секторе Всемирного банка. | Chima, Director of the UNDP Division for **Scientific** Development and **International** Affairs, and Ms. Cheryl Gray, Head of the Management Group in the **US** sector of the World Bank. | Чима, Директор Отдела по развитию научных кадров и международным вопросам ПРООН, и г-жа Шерил Грей, руководитель Группы по вопросам управления в американском секторе Всемирного банка. |

**Tabla 3.7:** Ejemplos de textos generados obtenidos al sustituir palabras empleando la técnica mixta.

| **Estadística** | **Valor** |
|---|---|
| Cantidad de frases originales | 1M |
| Cantidad de palabras nuevas | 2M |
| Cantidad de frases nuevas | 504K |
| Porcentaje de frases nuevas | 50,39 % |
| Media de palabras generadas por frase | 4,03 |
| Tiempo empleado | 11 días |

**Tabla 3.8:** Estadísticas recogidas de la generación de nuevos datos por sustitución de random empleando el modelo de lenguaje pre-entrenado *ruRoBERTa-large*. La M hace referencia a los millones y la K a los miles.

# CAPÍTULO 4
# Marco experimental

En el siguiente capítulo se va a presentar tanto la metodología como las tecnologías empleadas para el desarrollo de los modelos de traducción para evaluar la utilidad de las distintas técnicas de aumento de datos explicadas en la sección 3.2. Además, se van a entrenar dos modelos de traducción automática por cada técnica de aumento de datos, puesto que se quiere observar en que dirección de traducción se contempla mayor mejora.

## 4.1 Tecnologías empleadas

El lenguaje de programación principal que se ha empleado en el presente trabajo ha sido *Python* debido a que es un lenguaje dinámico, cuyo objetivo es alcanzar un código legible y fácil de entender. Además, posee una gran variedad de librerías que facilitan el trabajo en el ámbito del procesamiento del lenguaje natural y del aprendizaje automático.

Por otro lado, se ha empleado, aunque en menor medida, el lenguaje de programación empleado en la terminal y *shell* de Unix, *Bash*, debido a que a lo largo del desarrollo del trabajo se ha empleado el sistemo operativo Ubuntu 20.04. Por ello, se ha utilizado para la ejecución del código en *Python* que se ha ido desarrollando en el presente trabajo como para la ejecución de comandos en la propia terminal de Ubuntu. Además, *Bash* se ha empleado para la ejecución del código de la librería de traducción automática empleado en el proyecto.

#### OpenNMT-py

El desarrollo de los modelos de traducción se ha realizado empleando la herramienta *OpenNMT-py*[1] que es la versión en *pyTorch*[2] de la librería *OpenNMT*[3] el cual es una herramienta de código abierto empleado, en Traducción Automática Neuronal y Aprendizaje Neuronal Secuencial. Este proyecto fue llevado a cabo por el grupo de investigación de NLP de la Universidad de Harvard[4], SYSTRAN[5], lleva en marcha desde diciembre del 2016, ha sido empleado en varios proyectos de investigación y ha tenido un gran impacto en la industria [22].

Las principales ventajas que presenta esta librería y por lo que ha sido elegida para el desarrollo de los modelos de traducción son:

[1] https://github.com/OpenNMT/OpenNMT-py
[2] https://pytorch.org/
[3] https://opennmt.net/
[4] https://nlp.seas.harvard.edu/
[5] https://www.systransoft.com/

- Arquitecturas de modelos y procedimientos de formación altamente configurables.
- Capacidades eficientes de servicio de modelos para su uso en aplicaciones del mundo real.
- Extensiones para permitir otras tareas como generación de texto, etiquetado, resumen, imagen a texto y voz a texto.

## 4.2 Evaluación

Tras el proceso de entrenamiento de los modelos se tiene que realizar una evaluación de los mismos con el objetivo de valorar la calidad de las traducciones generadas. En el presente proyecto, se emplean cinco métricas automáticas BLEU [28], TER [35], chrF [30], METEOR [17] y NIST [12].

### 4.2.1. BLEU

*BiLingual Evaluation Understudy* (BLEU) [28], calcula la media geométrica de la precisión modificada por n-gramas, $p_n$, la cual consiste en el conjunto de todos los recuentos de n-gramas candidatos y sus correspondientes recuentos máximos de referencia, dichos recuentos de candidatos se recortan por su valor máximo de referencia correspondiente, sumado, y dividido por el número total de n-gramas candidatos, y multiplicado por el factor $BP$ que penaliza las frases cortas. Así pues, BLEU se puede modelizar de la siguiente manera:

$$BLEU = BP * exp\left(\sum_{n=1}^{N} \frac{log\ p_n}{N}\right) \tag{4.1}$$

La definición más común de BLEU se calcula sobre la concatenación de todas las frases de test, y normalmente emplea n-gramas de orden 4. Por otro lado, el resultado final de la métrica es un valor entre 0 y 1, a mayor valor mejor es la traducción. Dicho valor suele multiplicarse por 100 para obtener una mayor capacidad de interpretación representando el porcentaje.

### 4.2.2. TER

*Translation Error Rate* (TER) [35] esta basado en la alineación óptima, en términos de distancia de edición, de palabras de la frase de hipótesis y las palabras de la frase de referencia. Cada alineamiento consiste en una serie de transformaciones las cuales transforman la frase de hipótesis en la frase de referencia. Cada una de las transformaciones tiene asociada un coste. En consecuencia, la distancia de edición para un alineamiento se define como la suma de los costes de cada operación de transformación que se haya realizado en el alineamiento.

TER se define como el mínimo número de transformaciones necesarias para convertir una frase dada como hipótesis, en una de las frases dadas como referencias, normalizando dicha cantidad de transformaciones por la media de palabras de las referencias. Así pues, TER se puede formular de la siguiente manera:

$$TER = \frac{\#\ transformaciones}{\#\ palabras\ referencia} \tag{4.2}$$

| | |
|---|---|
| **REF** | ***Saudi Arabia*** denied ***this week*** information published in the ***American*** new york times |
| **HYP** | ***this week the Saudis*** denied information published in the new york times |

**Tabla 4.1:** Arriba, REF, frase de referencia, y abajo, HYP, frase de hipótesis empleadas para el cálculo de la métrica TER.

dichas transformaciones engloban la inserción, eliminación y sustitución o intercambio de palabras en las frase. Además, todas las transformaciones tienen asociado el mismo coste. En la tabla 4.1 se puede observar un ejemplo del cálculo de esta métrica, donde se remarcan en negrita y cursiva las diferencias entre la frases de referencia e hipótesis.

Como se puede observar, en dicha tabla la frase referente a la hipótesis es fluida y que además, ha falta de la palabra ***american***, tiene el mismo significado que la frase de referencia. Sin embargo, al calcular el valor de TER, no se considera dicha similitud, en cuanto a concordancia se refiere. Así, se puede ver que la frase ***this week*** en la frase de hipótesis se encuentra en una posición distinta respecto a la referencia. Por otra parte, la frase ***Saudi Arabia*** en la referencia aparece como ***the Saudis***, lo cual cuenta como dos sustituciones distintas. Y, por último, la palabra ***American*** aparece únicamente en la referencia.

A continuación, se va a aplicar el cálculo de TER sobre la referencia e hipótesis de la tabla 4.1. Así, se tiene que el número total de ediciones es 4, las cuales engloban, un cambio de posición, dos sustituciones y una inserción, y teniendo que el total de palabras de referencia es 13, se puede calcular el valor de TER como:

$$TER = \frac{4}{13} = 31\,\% \tag{4.3}$$

### 4.2.3. chrF

La métrica *chrF* [30] ha demostrado tener un potencial relevante para ser considerada como métrica independiente ya que tiene en cuenta algunos fenómenos morfosintácticos . Además, está basada en métricas como BLEU (ver 4.2.1) y TER (ver 4.2.2), ampliamente empleadas, son más complejas, mientras *chrF* es simple y, además, no requiere de ninguna herramienta adicional y/o fuentes de conocimiento, siendo también independiente del idioma y de la tokenización que se emplee.

El cálculo de la métrica viene dado por la siguiente ecuación:

$$chrF\ \beta = \left(1 + \beta^2\right) \frac{chrP \cdot chrR}{\beta^2 \cdot chrP + chrR} \tag{4.4}$$

donde chrP y chrR, hacen referencia a la *Precision* y el *Recall* de un caracter del *n-gram*, lo cuales han sido aritméticamente promediados sobre todos los *n-gram*, concretamente:

- *chrP* es el porcentaje de *n-gram* en la hipótesis que tiene su homólogo en la referencia.
- *chrR* es el porcentaje de caracteres de *n-grams* en la referencia que además aparecen en la hipótesis.

por otra parte, $\beta$ es un parámetro que da $\beta$ veces más importancia al *recall* que a la *precision*. Así, si tenemos una $\beta$ = 1, el *recall* y la *precision* tienen la misma importancia en el cálculo de chrF.

### 4.2.4. COMET

*Crosslingual Optimized Metric for Evaluation of Translation* COMET [33] es un *framework* basado en la librería *PyTorch* para el entrenamiento de modelos especializados en la evaluación de la calidad de los modelos de traducción, consiguiendo resultados de estado de arte correlación con el juicio humano. Así, aprovecha los recientes avances en el lenguaje multilingüe modelado para generar predicciones cercanas al juicio humano como *Direct Assessments*, *Human-mediated Translation Edit Rate* y métricas compatibles con el *framework* de *Multidimensional Quality Metric*.

Una de las novedades de esta métrica, es la incorpfrase del texto de entrada de la traducción, es decir, el texto de entrada, en vez de emplear únicamente la traducción de referencia y la frase de hipótesis generada por el modelo de traducción automática.

En cuanto a la arquitectura los distintos modelos que se han entrenado en el desarrollo de COMET, todos ellos contienen una primera capa que ha llamado, *Cross-lingual Encoder* la cual es una capa de codificador (ver 2.2.2) que emplea el modelo base de XLM-R [7]. Así, dada una frase de entrada $\mathbf{x} = (x_0, x_1, \ldots, x_n)$ el codificador produce un *embedding* $e_j^{(\ell)}$ para cada token $x_j$ y para cada capa $\ell \in \{0, 1, \ldots, k\}$. Dicho proceso se aplica a la secuencia de entrada, la referencia y la hipótesis, con el objetivo de mapear dichas secuencias en un espacio de características compartido. Por otro lado, puesto que se ha demostrado que, a menudo, si se coge únicamente los *embeddings* de la última capa esto conlleva a lograr un rendimiento inferior, debido a que distintas capas pueden conseguir diferentes niveles de correlación. Por ello, se añade una segunda capa denominada *Pooling Layer* la cual reúne los *embeddings* de las capas del codificador más relevantes, en un único *embedding* para cada token, $e_j$ empleando un mecanismo de atención (ver 2.2.3). Para el cálculo de dicho *embedding*, se emplea la siguiente ecuación:

$$e_{x_j} = \mu \mathbf{e}_{x_j}^{\top} \boldsymbol{\alpha} \tag{4.5}$$

donde, $\mu$ es el peso del coeficiente de entrenamiento, $\mathbf{e}_j = [e_j^{(0)}, e_j^{(1)}, \ldots, e_j^{(k)}]$ es el vector de la capa de *embeddings* para el token $x_j$ y, por último, $\boldsymbol{\alpha} = softmax([\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(k)}])$ es el vector correspondiente a los pesos de las capas entrenables. Además, con el objetivo de evitar posibles sobreajustes de la información contenida en cada capa individual, se ha empleado un *dropout* en el cual con una probabilidad de $p$, el peso $\alpha^{(i)}$ se vuelca a $-\infty$.

En las figura 4.1 se puede observar la arquitectura del modelo estimador. La frase de entrada (*Source*), la frase de referencia (*Reference*) y la frase de hipótesis (*Hypothesis*) se codifican de manera independiente una de otra, mediante el uso de un codificador preentrenado. A continuación, el resultante *embedding* a nivel de palabra se pasa a través de la denominada *Pool Layer* para crear el *embedding* a nivel de frase para cada segmento. Por último, el resultantes *embeddings* son combinados y concatenados en un único vector por una capa *feed-forward* con regresión. El modelo es entrenado con el objetivo de minimizar el Error Cuadrático Medio o en inglés *Mean Squared Error* (MSE).

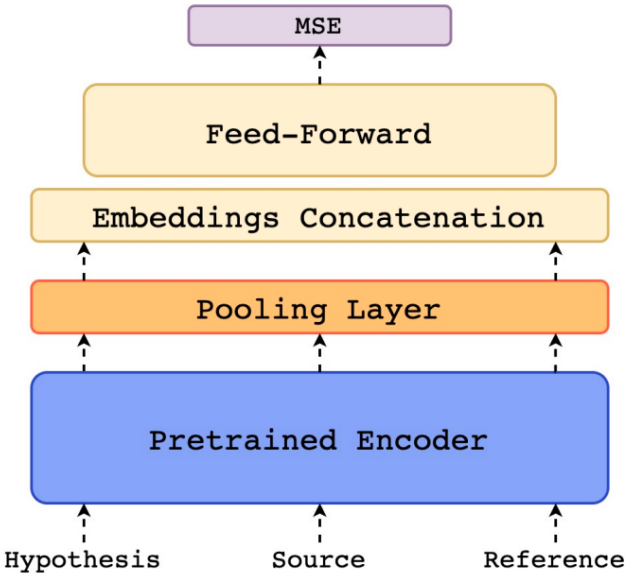


**Figura 4.1:** Aquitectura del modelo estimador de COMET [33].

### 4.2.5. BEER

BEER (*Better Evaluation as Ranking*) [38] es un modelo de evaluación el cual es una interpolación linear simple de funciones de características siendo, además, sencillo de entrenar y fácil de interpretar. El valor de la función lineal viene dado por la siguiente ecuación:

$$score(h,r) = \sum_i w_i \times \phi_i\,(h,r) \tag{4.6}$$

donde, $h$ es la hipótesis, $r$ es la referencia, $w_i$ el peso asignado a cada característica $\phi_i\,(h,r)$. Así, el modelo determina la similitud existente entre $h$ y $r$ mediante la asignación de $w_i$ a cada $\phi_i\,(h,r)$.

### 4.2.6. NIST

El nombre de NIST [12] proviene de *National Institute of Standards and Technology* que fue la institución que desarrolló la métrica. Así, dicha métrica está basada en el cálculo de BLEU (ver 4.2.1), con ciertas modificaciones. Además de calcular la precisión de los n-gramas añadiendo pesos iguales a cada uno, NIST añade cuán informativo es un n-grama particular. La información de los pesos se cómputa empleando el conteo de *n-grams* sobre el conjunto de referencias, como se puede observar en la siguiente ecuación:

$$I\Big(w_i,\ldots,w_n\Big) = \log_2\left(\frac{\#\ ocurrencias\ de\ w_1,\ldots,w_{n-1}}{\#\ ocurrencias\ de\ w_1,\ldots,w_n}\right) \tag{4.7}$$

Por lo tanto, teniendo en cuenta los resultados obtenidos de los F-ratios del conteo de la información de los pesos y la comparativa de las correlaciones, se propone una

modificación en el cálculo del BLEU que da lugar a la siguiente fórmula que define el cálculo de NIST:

$$score(h,r) = \sum_{n=1}^{N} \left\{ \sum_{w_i,\ldots,w_n} I\left(w_i,\ldots,w_n\right) / \sum_{w_i,\ldots,w_n \ in \ h} \left(1\right) \right\} \cdot \exp\left\{ \beta \log^2 \left[ \text{mín}\left( \frac{L_h}{\overline{L_r}}, 1 \right) \right] \right\} \tag{4.8}$$

donde, $\beta$ se escoge para realizar el factor de penalización de brevedad $= 0{,}5$ cuando el número de palabras en la salida de la hipótesis es de dos tercios de la media del número de palabras en la referencia. Además:

- $\overline{L_r}$ es la media de la cantidad de palabras de la referencia, promediado sobre todas las referencias.
- $L_h$ es la cantidad de palabras de la hipótesis que han obtenido una puntuación.

## 4.3 Conjunto de datos

En esta sección, se introduce el conjuntos de datos que se ha utilizado a lo largo del desarrollo del presente proyecto. Dichos conjuntos de datos están estructurado a nivel de documento. Además, se han tokenizado y limpiado mediante varios métodos de preprocesado de datos. Por último, cabe destacar que el conjunto de datos empleado en este proyecto proviene de la Organización de las Naciones Unidas (ver 4.3.1) el cual se va a describir a continuación.

### 4.3.1. ONU

El *United Nations Parallel Corpus*[6] es el corpus paralelo de las Naciones Unidas y está compuesto por documentos parlamentarios y registros oficiales de las Naciones Unidas que se encuentran disponibles al dominio público. Dicho conjunto de datos contiene actualmente contenido que fue generado y traducido manualmente entre 1990 y 2014, incluyendo alineamientos a nivel de frase. Aunque este conjunto de datos está disponible en las seis lenguas oficiales de las Naciones Unidas, para el presente trabajo solo se ha empleado el conjunto de datos paralelo Inglés-Ruso.

Por último, cabe destacar que la creación del conjunto de datos fue impulsada por el compromiso de la ONU con el multilingüismo y como reacción a la creciente importancia de las traducción automática dentro de los servicios de traducción del Departamento de la Asamblea General y Gestión de Conferencias (DGACM) y el sistema SMT de la ONU, Tapta4UN [13].

#### UN Parallel Corpus inglés-ruso

En las tablas 4.2, 4.3 y 4.4 se muestran las estadísticas pertenecientes al conjunto de datos paralelos de las Naciones Unidas para la combinación de idiomas inglés-ruso y estadísticas generales que engloban todas las lenguas del conjunto de datos.

El número de tokens fue calculado tras aplicar el preproceso de tokenización con Moses [24]

[6] https://conferences.unite.un.org/UNCORPUS/#introduction

| Estadística | Inglés | Ruso |
|---|---|---|
| Número de tokens | 601M | 570M |
| Número de documentos | 133K | 133K |
| Número de líneas | 23M | 23M |

**Tabla 4.2:** Estadísticas recogidas del conjunto de datos de las Naciones Unidas para la combinación de idiomas inglés-ruso. La M hace referencia a los millones y la K a los miles.

| Documentos total | Pares de documentos alineados |
|---|---|
| 779K | 2M |

**Tabla 4.3:** Estadísticas de los documentos contenidos en el conjunto de datos de las Naciones Unidas. La M hace referencia a los millones y la K a los miles.

| Número de documentos | Número de líneas | Número de tokens en inglés |
|---|---|---|
| 86K | 11M | 335M |

**Tabla 4.4:** Estadísticas de los subconjuntos de datos totalmente alineados en el conjunto de datos de las Naciones Unidas. La M hace referencia a los millones y la K a los miles.

## 4.4 Preparación de los datos

El conjunto de datos que se ha empleado en el presente trabajo ha pasado por un proceso de limpieza creado específicamente teniendo en cuenta las peculiaridades de ambos idiomas y las necesidades establecidas por el modelo de generación de datos y de la calidad que se pretende lograr con el presente trabajo. El objetivo de dicho proceso es crear un conjunto de datos final el cual ha superado ciertos criterios de calidad para así lograr obtener el conjunto de datos más relevante y representativo posible a partir de los datos originales. Por un lado, tenemos el conjunto de métodos que conforman la normalización de un segmento, al cual llamaremos **normalizadores**. Por otra parte, se define el conjunto de métodos que determinan si un texto es válido, al cual llamaremos **validadores**. A continuación, se explicarán cada uno de los métodos de los dos conjuntos.

### 4.4.1. Normalizadores

El objetivo de estás técnicas de normalización es lograr que los segmentos que contienen algún tipo de ruido corregible, se puedan seguir empleando aplicándoles dichas correciones. A continuación, se muestra el listado de normalizadores que se han implementado en el presente proyecto:

- Normalización de las puntuaciones repetidas a una puntuación, por ejemplo, se pasa de !!! a tener !, tanto en la frase origen como en la destino.
- Normalización de espacios en blanco repetidos a un único espacio en blanco, tanto en la frase origen como en la frase destino.

### 4.4.2. Validadores

El objetivo de estas técnicas de validación es determinar si un segmento es lo suficientemente aceptable para poder pertenecer al conjunto de datos finales. A continuación, se muestra el listado de validadores que se han implementado en el presente proyecto:

- Eliminación de segmentos que no cumplan cierto rango de longitud, debido a que el codificador de BERT acepta como máximo 512 tokens.
- Eliminación de segmentos que contengan en su mayoría números, puesto que dichos segmentos no contienen información relevante para la generación de nuevos segmentos a partir de dichos segmentos.
- Eliminación de segmentos vacíos.
- Eliminación de segmentos que sean iguales en el lado ruso y en el lado inglés.
- Eliminación de segmentos que sólo contengan direcciones de correos electrónicos o de Internet.
- Eliminación de segmentos que, en el lado del ruso, contengan más cantidad de elementos en el alfabeto latino que en el alfabeto cirílico.
- Eliminación de segmentos que no contengan en el lado ruso y en el lado inglés los mismos y la misma cantidad de números y elementos.

## 4.5 Creación de los conjuntos de datos

En la presente sección, se va a exponer la partición de los distintos conjuntos de datos que se ha realizado, los cuales se han empleado en el entrenamiento, validación y testeo de los distintos modelos de traducción automática entrenados en el proyecto.

Por lo tanto, dicha partición se ha realizado en tres conjuntos de datos bien diferenciados: entrenamiento, validación y test. Además, para la generación conjunto de test, se han aplicado dos técnicas de aumentos de datos, la técnica de sustitución mixta, presentada en el apartado 3.2.4, con el que se consiguió obtener cerca de un 35 % de frases nuevas en ruso, y la técnica de sustitución de sinónimos, presentada en el apartado 2.4.3, con el que se consiguió, gracias a Natalia Loukachevitch (*louk_nat@mail.ru*) que fue quién proporciono el *theshaurus*[7], es decir, el diccionario de sinónimos de ruso, alrededor de un 13 % de frases nuevas. Por último, se hizo una mezcla de todos datos de test recogidos, se seleccionaron 2000 muestras y se generó mediante un modelo de traducción automática la parte del inglés, generando así, el conjunto de test final, el cual se ha empleado para extraer las métricas automáticas que se comentan en las secciones posteriores. En la Tabla 4.5 se pueden observar las estadísticas recogidas de los diferentes conjuntos de entrenamiento y validación, y la representación de como se ha generado el conjunto de test.

[7] https://ruwordnet.ru/en

| Conjunto | # Muestras |
|---|---|
| **Entrenamiento** | |
| Original | 7.9M |
| Adverbios | 271K |
| Sustantivos | 916K |
| Adjetivos | 792K |
| Mixta | 501K |
| **Validación** | 2K |
| **Test** | |
| Original | 2K |
| Aumentado + Mixta | 2.7K |
| Aumentado + sinónimos | 2.3K |
| Final | 2K |

**Tabla 4.5:** Estadísticas del conjunto de datos original y lo distintos conjuntos de datos generados empleando la técnicas de sustitución de adverbios, sustantivos, adjetivos y mixta, para el par de idiomas inglés-ruso. Además, se presenta como se ha generado el test. M representa millones y K representa miles.

## 4.6 Experimentación

Con el objetivo de corroborar que la adición de un conjunto de datos que se genera a partir del conjunto de datos de entrenamiento original permite que el modelo de traducción proporcione traducciones con una mayor calidad que el sistema original o *baseline* que únicamente se ha entrenado con el conjunto de datos inicial, se va a realizar una experimentación con los diferentes conjuntos de datos que se han generado y expuesto en el presente proyecto (ver 3.2).

Por lo tanto, una vez preparados los datos de entrenamiento y separados creados los modelos de traducción automática, el paso siguiente es entrenar dichos modelos. Para ello, es necesaria la especificación de los parámetros asociados a los sistemas de traducción. En primer lugar, se define como modelo *baseline* y como modelos para el reentrenamiento con los datos generados, los parámetros de los sistemas neuronales del modelo *Transformer*, a nivel de frase, compuesto por 6 capas tanto de codificador como decodificador, ambos con un tamaño de 512, una dimensionalidad de 2048 unidades en las capas internas, de 8 cabezales de autoatención, una longitud de las frases de entrada y salida configurada en 400 y con un ratio de *dropout* de 0.1 [37].

En cuanto al entrenamiento, se emplea un tamaño de *batch* de 4096 tokens el cual indica la frecuencia con la que el gradiente se actualiza, y un total de 100K pasos de entrenamiento, que señalan la cantidad de veces que se entrena el modelo sobre los datos proporcionados. Además, se emplea el optimizador SGD [20] con un $\epsilon = 0.05$ que hace referencia al ratio de aprendizaje empleado, el método de *decay* noam y, por último, se han empleado los 8k primeros pasos de entrenamiento como pasos de calentamiento del modelo de traducción automática.

La arquitectura comentada, se emplea tanto para los entrenamientos de los modelos de traducción tanto del ruso al inglés como, del inglés al ruso. Sin embargo, para el uso del conjunto de datos generado, se ha empleado la técnica de reentrenamiento a partir del último paso de entrenamiento de los modelos *baseline* y hasta los 150K pasos de entrenamiento, es decir, se realizan 50K pasos de entrenamiento extra.

## 4.7 Hardware empleado

En el presente proyecto se han entrenado los diferentes modelos en una GPU TITAN Xp. Dependiendo del conjunto de datos para el que se entrene los modelos la duración del entrenamiento varía.

En relación con el conjunto de datos de la ONU, cuyo entrenamiento conlleva 100K pasos de entrenamiento y con los pares de idiomas inglés-ruso y ruso-inglés, cada época de entrenamiento tarda alrededor de 63 segundos, comprendiendo un entrenamiento total de 17,5 horas. Por otro lado, los modelos que han sido entrenados con el conjunto de datos generado por las distintas técnicas de aumento de datos y que han conllevado 50K pasos de entrenamiento adicionales, partiendo del modelo de 100K de su respectivo *baseline* para los pares de idiomas inglés-ruso y ruso-inglés, cada época tarda alrededor de 54 segundos, comprendiendo un entrenamiento de 15 horas.

# CAPÍTULO 5
# Resultados experimentales

En este apartado del presente trabajo se van a exponer los resultados obtenidos de los distintos modelos de traducción automática que se han entrenado. Dichos modelos, se van a evaluar mediante seis métricas de evaluación de traducción automática, escogidas minuciosamente por su gran capacidad de evaluación, que hasta la fecha son las que más se acercan a la evaluación humana.

## 5.1 Análisis cuantitativo

En esta sección, se va a presentar el análisis cualitativo, cuyo objetivo es evaluar de forma automática mediante el empleo de métricas de evaluación automática, la correlación entre los textos generados por los modelos de traducción automática entrenados en el presente proyecto y el conjunto de test comentado en la sección 4.5, y los resultados de los cuales se puede ver en la tablas 5.1 y 5.2. Para ello, se distingue entre los modelos de traducción automática inglés-ruso y ruso-inglés. Además, cabe destacar que se han realizado cálculos de los intervalos de confianza para los valores obtenidos de BLEU y TER mediante el empleo de la técnica *bootstrap resampling*[1] [3], la cual consiste en volver a muestrear nuestra muestra original con reemplazo (muestra de *Bootstrap*) y generar réplicas de *Bootstrap* mediante el uso de estadísticas de resumen. Por último, debido a que la técnica que se ha empleado para calcular los intervalos de confianza no es lo suficientemente significativa desde un punto de vista estadístico, se ha realizado el cálculo *approximate randomization test*[2] entre el modelo *baseline* y el resto de modelos entrenados durante el desarrollo del proyecto, con el objetivo de determinar si los modelos entrenados son estadísticamente diferentes al *baseline*, es decir, el valor de BLEU es lo suficientemente significativo para determinar cual de los modelos comparados es mejor.

### 5.1.1. Resultados inglés-ruso

En primer lugar, se van a presentar los resultados obtenidos del análisis cuantitativo de los modelos entrenados en la dirección de traducción inglés-ruso. Así, se pueden observar en la tabla 5.1 los resultados obtenidos al aplicar las distintas métricas de evaluación de los distintos modelos de traducción entrenados. Además se puede observar los intervalos de confianza que se han calculado para las métricas de BLEU y TER.

A la vista de los resultados obtenidos en la tabla 5.1, se puede observar que el modelo *baseline* no solapa, tanto en el valor de BLEU como en el de TER, con ningún modelo

[1] https://github.com/midobal/mt-scripts/tree/master/confinter
[2] https://github.com/midobal/mt-scripts/tree/master/art

**EN-RU**

| Técnica de sustitución | BLEU | TER | CHRF-2 | NIST | BEER | COMET |
|---|---|---|---|---|---|---|
| Baseline | 58.4 ± 1.4 | 30.9 ± 1.1 | 78.4 | 2.9 | 74.1 | 90.6 |
| Adverbios | 69.3 ± 1.4 | 21.4 ± 1.0 | 84.6 | 3.2 | 79.3 | 97.8 |
| Sustantivos | 68.3 ± 1.4 | 21.7 ± 1.0 | 84.0 | 3.2 | 80.0 | 99.9 |
| Adjetivos | 67.9 ± 1.4 | 22.6 ± 1.0 | 83.6 | 3.2 | 78.8 | 97.2 |
| Mixta | 69.8 ± 1.3 | 20.9 ± 1.0 | 84.7 | 3.2 | 80.4 | 99.6 |

**Tabla 5.1:** Resultados obtenidos a la hora de evaluar las técnicas de aumento de datos por sustitución de adverbios, sustantivos, adjetivos y mixta, mediante las métricas de evaluación BLEU, TER, chrF-2, NIST, BEER y COMET, en los modelos de traducción del inglés al ruso. El valor que sigue al elemento ± hace referencia al intervalo de confianza.

entrenado con un conjunto de datos aumentado, lo que nos viene a decir que, con un 95 % de seguridad, los dos modelos que se comparan, en este caso el *baseline* con cualquier otro, son significativamente distintos, por lo que se puede afirmar que los modelos entrenados con su respectivo conjunto de datos aumentado, son mejores que el modelo *baseline*. Por otro lado, comparando los modelos que han sido entrenados con las técnicas de aumento de datos, se puede apreciar que para cada par de modelos que se escojan, sus valores en el intervalo de confianza de BLEU y TER solapan entre sí, por lo tanto, no se puede saber con total seguridad como de distintos son los pares de modelos entre sí, pero se puede intuir cierta tendencia del modelo entrenado con el conjunto de datos obtenidos al emplear la técnica de sustitución mixta, a ser mejor en cuanto a las métricas de evaluación automática empleadas, que el resto de modelos entrenados. Por último, se han comparado, mediante la técnica de *approximate randomization test* el modelo *baseline* con el resto de modelos, dando como resultado en todos los casos un valor inferior de *p-value* el cual esta establecido en 0.05 y lo que nos viene a decir que los modelos comparados son estadísticamente diferentes, lo que nos viene a decir, al igual que en el caso de los intervalos de confianza, que todos los modelos comparados con el *baseline* mejoran.

### 5.1.2. Resultados ruso-inglés

En segundo lugar, se van a presentar los resultados obtenidos del análisis cuantitativo de los modelos entrenados en la dirección de traducción ruso-inglés. Así, se pueden observar en la tabla 5.2 los resultados obtenidos al aplicar las diversas métricas de evaluación, descritas en la sección 4.2, a los resultados obtenidos al traducir el conjunto de test, empleando los distintos modelos de traducción entrenados. Además se puede observar los intervalos de confianza que se han calculado para las métricas de BLEU y TER.

**RU-EN**

| Técnica de sustitución | BLEU | TER | CHRF-2 | NIST | BEER | COMET |
|---|---|---|---|---|---|---|
| Baseline | 39.4 ± 1.1 | 46.7 ± 1.1 | 61.7 | 2.6 | 62.9 | 26.4 |
| Adverbios | 59.3 ± 1.3 | 30.9 ± 1.1 | 75.0 | 3.0 | 72.2 | 24.3 |
| Sustantivos | 57.6 ± 1.2 | 32.1 ± 1.1 | 75.7 | 3.0 | 73.0 | 22.4 |
| Adjetivos | 60.0 ± 1.3 | 30.5 ± 1.1 | 75.5 | 3.1 | 72.9 | 26.0 |
| Mixta | 59.3 ± 1.3 | 31.4 ± 1.1 | 75.7 | 3.1 | 73.0 | 23.0 |

**Tabla 5.2:** Resultados obtenidos aplicando las técnicas de aumento de datos por sustitución de adverbios, sustantivos, adjetivos y mixta, para las métricas de evaluación BLEU, TER, chrF-2, NIST, BEER y COMET, en los modelos de traducción automática de dirección ruso-inglés. El valor que sigue al elemento ± hace referencia al intervalo de confianza.

A la vista de los resultados obtenidos en la tabla 5.2, se puede observar que el modelo *baseline* no solapa, tanto en el valor de BLEU como en el de TER, con ningún modelo

entrenado con un conjunto de datos aumentado, lo que nos viene a decir que, con un 95 % de seguridad, los dos modelos que se comparan, en este caso el *baseline* con cualquier otro, son significativamente distintos, por lo que se puede afirmar que los modelos entrenados con su respectivo conjunto de datos aumentado, son mejores que el modelo *baseline*. Por otro lado, comparando los modelos que han sido entrenados con las técnicas de aumento de datos, se puede apreciar que para cada par de dichos modelos que se escojan, sus valores en el intervalo de confianza de BLEU y TER solapan entre sí, por lo tanto, no se puede saber con total seguridad como de distintos son los pares de modelos entre sí, sin embargo, se puede observar que para los pares de modelos de sustantivos-adjetivos y sustantivos-mixta, prácticamente no solapan, por lo que se puede intuir que la tendencia es que tanto el modelo de sustitución de adjetivos y mixta, pueden ser mejores que el modelo de sustitución de sustantivos, predominando el modelo de sustitución de adjetivos. Por último, para el cálculo de *approximate randomization test*, en todos los pares de comparación del modelo *baseline* y el cada uno de los modelos entrenados con su conjunto de datos aumentado, se consigue un valor inferior al *p-value*, por lo que se puede afirmar que cada uno de los modelos de traducción automática entrenado con su respectivo conjunto de datos aumentado, son estadísticamente distintos al modelo *baseline* y, por tanto, los valores de BLEU se pueden considerar relevantes para afirmar que hay mejora en todos los modelos frente al modelo *baseline*.

Cabe destacar, que en comparación con los resultados cuantitativos obtenidos en el apartado 5.1.1, los resultados obtenidos en este apartado son significativamente peores en cuanto a las métricas de evaluación. Esto es debido a que el conjunto de datos de inglés ha sido generado mediante un modelo de traducción automática, a partir de los datos generados por cada técnica de aumento de datos empleada en el presente proyecto. Ello conlleva, que la calidad de la traducción es peor que en el caso del ruso, en el que se han hecho cambios a nivel de palabra y, además, siguiendo criterios de preservación de la calidad en la medida de lo posible a la hora de generar el nuevo conjunto de datos.

## 5.2 Análisis cualitativo

En la siguiente sección, se va a presentar el análisis cualitativo que se ha realizado de los modelos de traducción automática entrenados en las dos direcciones de traducción, inglés-ruso y ruso-inglés. Para exponer el análisis cualitativo de los distintos modelos de traducción, se van a presentar ejemplos de traducciones que han realizado cada modelo de traducción

### 5.2.1. Resultados inglés-ruso

| Técnica de sustitución | Ejemplo |
|---|---|
| Origen en inglés | Their case also attracted the attention of the control commission in Geneva. |
| Original en ruso | Их дело также привлекло внимание комиссии по контролю в Женеве. |
| Adverbios | Их случай также привлек внимание контрольной комиссии в Женеве. |
| Sustantivos | Их дело также привлекло внимание комиссии по контролю в Женеве. |
| Adjetivos | Их дело также привлекло внимание комиссии по контролю в Женеве. |
| Mixta | Их дело также привлекло внимание контрольной комиссии в Женеве. |

**Tabla 5.3:** Ejemplo de traducción automática obtenidos a la hora de traducir empleando los modelos entrenados con los conjuntos de datos generados por las técnicas de aumento de datos por sustitución de adverbios, sustantivos, adjetivos y mixta, escogidos de forma aleatoria donde la frase de entrada en inglés y las salidas en ruso.

En la tabla 5.3, se puede observar un ejemplo de traducción de una frase en inglés a las distintas traducciones a ruso empleando cada uno de los modelos de traducción automática entrenados en dicha dirección. Así, si se compara la traducción generada por el modelo *baseline* con el resto de modelos de traducción, se tiene:

- En el caso Baseline-Adverbios, se puede observar que hay una significativa diferencia entre ambas traducciones, donde la primera diferencia se puede interpretar como un cambio de sinónimo y las dos siguientes son adaptaciones a la primera palabra para que la frase tenga coherencia.
- En el caso Baseline-Sustantivos y Baseline-Adjetivos, no hay cambios frente a la traducción generada por el modelo *baseline*
- En el caso Baseline-Mixta, hay un cambio de analogía, es decir, se pasa de tener комиссии по контролю a контрольной комиссии que viene a ser una reestructuración de la frase.

### 5.2.2. Resultados ruso-inglés

| Técnica de sustitución | Ejemplo |
|---|---|
| Origen en ruso | Их случай также привлек внимание комиссии по контролю в Женеве. |
| Original en inglés | **The case** also **brought** to the attention of the Geneva **Monitoring Commission** |
| Adverbios | The **other case** also **drew** the attention of the **control commission** in Geneva. |
| Sustantivos | The case also **attracted** the attention of the **control commission** in Geneva. |
| Adjetivos | The case also **drew** the attention of the **control commission** in Geneva. |
| Mixta | **Others also drew** the attention of the **control commission** in Geneva. |

**Tabla 5.4:** Ejemplo de traducción automática obtenidos a la hora de traducir empleando los modelos entrenados con los conjuntos de datos generados por las técnicas de aumento de datos por sustitución de adverbios, sustantivos, adjetivos y mixta, escogidos de forma aleatoria donde la frase de entrada en ruso y las salidas en inglés.

En la tabla 5.4, se puede observar un ejemplo de traducción de una frase en ruso a las distintas traducciones a inglés empleando cada uno de los modelos de traducción automática entrenados en dicha dirección. Por lo tanto, si se compara la traducción generada por el modelo *baseline* con el resto de modelos de traducción:

- En el caso Baseline-Adverbios, se puede observar varias diferencias entre ambas traducciones. En primer lugar, se añade la palabra *other* antes de la palabra *case*. Además, hay dos cambios más, el primero es *brought* cambia a *drew* y *Monitoring Commission* a *control commission*.
- En el caso Baseline-Sustantivos, se observan menos diferencias en comparación con el caso Baseline-Adverbios, puesto que únicamente se realizan dos cambios, *Monitoring Commission* a *control commission*, igual que en el caso anterior, y *brought* cambia a *attracted*.
- En el caso Baseline-Adjetivos, sucede algo similar al caso Baseline-Sustativos, donde cambia *brought* por *drew* y *Monitoring Commission* a *control commission*.
- En el caso Baseline-Mixta, persiste la diferencia de *Monitoring Commission* a *control commission* como en los casos anteriores y, además, se cambia *The case* a *Others also* y *brought* a *drew*.

# CAPÍTULO 6
# Conclusiones

En este último apartado del proyecto, se van a exponer la evaluación de los objetivos que se propusieron al principio del escrito. Además, se va a definir las distintas líneas que se podrían seguir para el desarrollo futuro del presente trabajo.

## 6.1 Evaluación de los objetivos

En el presente proyecto, se han generado cuatro conjuntos de datos totalmente nuevos empleando un modelo de lenguaje pre-entrenado con el objetivo de mejorar los sistemas de traducción neuronal. Además, se ha observado la importancia que tiene una buena selección de palabras generadas sobre la generación final de los conjuntos de datos.

Dichos conjuntos de datos se han empleado para el entrenamiento de ocho modelos de traducción, cuatro en dirección inglés-ruso y cuatro en dirección ruso-inglés, obteniendo un resultados de mejora. Dichos resultados, muestran que los modelos re-entrenados con conjuntos de datos generados de forma automática mejoran las traducción en comparación con los modelos originales, entrenados únicamente con el conjunto de datos principal.

En conclusión, cabe destacar que los resultados muestran que emplear datos aumentados automáticamente para el entrenamiento de modelos de traducción son una solución viable cuando no se dispone de la suficiente cantidad y, además, ayudan a alcanzar mejores traducciones. Sin embargo, aún queda mucho trabajo que realizar en este campo.

## 6.2 Trabajo futuro

Una primera línea futura, podría ser añadir el método de sustitución de verbos en una frase. Dicha técnica, guarda cierta relación con las técnicas propuestas en este proyecto, pero no se ha llegado a implementar en el presente trabajo debido a su alta dificultad, puesto que, se tienen que tener en cuenta aspectos tanto del verbo como pueden ser el género, el tiempo verbal, el número y la persona, como de las palabras que se ven influenciadas por el verbo que se va a cambiar. Sin embargo, con ello se conseguiría una mayor variabilidad de los texto generados.

Por otra parte, otra línea futura sería el empleo de un sistema de alineación de palabras paralelo, el cual se emplearía para que al cambiar una palabra del texto de entrada, solo sería necesario realizar la traducción de dicha palabra al idioma del texto de salida, en la posición dada por el alineador. Con ello, se conseguiría no solo ahorrar en cuanto a recursos necesarios para llevar a cabo la traducción automática, sino que también en

tiempo de cómputo. Además, ello ayudaría a mitigar el ruido generado al emplear un traductor automático para la traducción de toda la frase.

Por último, una mejora respecto al sistema actual de aumento de datos, sería emplear computación en la nube para acelerar el proceso de creación de los conjuntos de datos aumentados y del entrenamiento de los modelos de traducción automática. Servicios como AWS (*Amazon Web Services*) Cloud[1], Azure Cloud[2] o Google Cloud[3] sería los candidatos a emplear para la realización de dicha mejora.

[1]https://aws.amazon.com/
[2]https://azure.microsoft.com/en-us/
[3]https://cloud.google.com/

# Bibliografía


[1] Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, 54:5789–5829, 2021.

[2] Dzmitry Bahdanau, Kyunghyun Cho, Université De Montréal, Yoshua Bengio, and Université De Montréal. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[3] Ahlem Bougarradh, Slim M'hiri, and Faouzi Ghorbel. Introduction of the bootstrap resampling in the generalized mixture estimation. In *3rd International Conference on Information and Communication Technologies: From Theory to Applications*, pages 1–6, 2008.

[4] Brown, Pietra, Pietra, and Mercer. A statistical approach to sense disambiguation in machine translation. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California*, 1991.

[5] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311, 1993.

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

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, 2020.

[8] Debajit Datta, Preetha Evangeline, Dhruv Mittal, and Anukriti Jain. Neural machine translation using recurrent neural network. *International Journal of Engineering and Advanced Technology*, 9:1395–1400, 2020.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Compu-*

*tational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019.

[11] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600, 2017.

[12] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, page 138–145, San Francisco, CA, USA, 2002.

[13] Cecilia Elizalde, Bruno Pouliquen, Christophe Mazenc, and José García-Verdugo. Tapta4un: collaboration on machine translation between the world intellectual property organization and the united nations. London, UK, 2012. Aslib.

[14] Marzieh Fadaee and Christof Monz. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium, 2018.

[15] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[16] Sahar Ghannay, Benoit Favre, Yannick Estève, and Nathalie Camelin. Word embedding evaluation and combination. *Language Resources and Evaluation*, pages 300–305, 2016.

[17] Lifeng Han. Machine translation evaluation resources and methods: A survey. In *IPRC- Irish Postgraduate Research Conference*, pages 5–21, Dublin, Ireland., 2018.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[19] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, 2020.

[20] Kenji Kawaguchi and Haihao Lu. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pages 669–679, 2020.

[21] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. In *ICLR 2017 conference submission*, 2017.

[22] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, 2017.

[23] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[24] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, 2007.

[25] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, 2019.

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

[27] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander Fraser, Shankar Kumar, Libin Shen, David A Smith, Katherine Eng, et al. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168, Boston, Massachusetts, USA, 2004.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002.

[29] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 11:1–8, 2017.

[30] Maja Popović. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, 2015.

[31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[32] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. page 9, 2019.

[33] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, 2020.

[34] Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. 1999.

[35] M. Snover, Bonnie J Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, 2006.

[36] Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic, 2021.

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[38] Miloš Stanojević and Khalil Sima'an. Beer: Better evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA, 2014.

[39] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pages 194–206. Springer International Publishing, 2019.

[40] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, volume 27, page 3104–3112, Cambridge, MA, USA, 2014.

[41] Ghulam Rasool Tahir, Sohail Asghar, and Nayyer Masood. Knowledge based machine translation. In *2010 International Conference on Information and Emerging Technologies*, volume 4, pages 5–24, 1989.

[42] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279, 2018.

[43] Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21, 2020.

[44] Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning, 2019.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017.

[46] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[47] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, 2019.

[48] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020.

[49] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, volume 32, 2019.

[50] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.

[51] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, 2021.

# APÉNDICE A
# OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

| **Objetivos de Desarrollo Sostenible** | **Alto** | **Medio** | **Bajo** | **No procede** |
|---|---|---|---|---|
| ODS 1. **Fin de la pobreza.** | | | | X |
| ODS 2. **Hambre cero.** | | | | X |
| ODS 3. **Salud y bienestar.** | | | | X |
| ODS 4. **Educación de calidad.** | X | | | |
| ODS 5. **Igualdad de género.** | | | X | |
| ODS 6. **Agua limpia y saneamiento.** | | | | X |
| ODS 7. **Energía asequible y no contaminante.** | | | | X |
| ODS 8. **Trabajo decente y crecimiento económico.** | X | | | |
| ODS 9. **Industria, innovación e infraestructuras.** | X | | | |
| ODS 10. **Reducción de las desigualdades.** | | X | | |
| ODS 11. **Ciudades y comunidades sostenibles.** | | | | X |
| ODS 12. **Producción y consumo responsables.** | | | | X |
| ODS 13. **Acción por el clima.** | | | | X |
| ODS 14. **Vida submarina.** | | | | X |
| ODS 15. **Vida de ecosistemas terrestres.** | | | | X |
| ODS 16. **Paz, justicia e instituciones sólidas.** | | X | | |
| ODS 17. **Alianzas para lograr objetivos.** | | X | | |

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

*El presente trabajo fin de grado está relacionado con varios de los objetivos de desarrollo sostenible (ODS). Desde el inicio de los tiempos, la comunicación ha supuesto un intercambio constante de conocimiento y recursos entre distintas sociedades, con el objetivo de socializar y realizar negociaciones, lo que ha supuesto el nacimiento de las sociedades primitivas y la estabilidad y avance de las sociedades de las posteriores. Sin embargo, hoy en día el objetivo principal es tener la capacidad de establecer una comunicación efectiva y sencilla teniendo en cuenta la enorme cantidad de diferentes idiomas y sus características propias. Por ello, la traducción cada vez juega un papel más y más en el desarrollo social, cultural y económico. Debido a esto se hace obvio la creciente necesidad de tener un acceso rápido, sencillo y globalizado a traducciones de calidad. Además, ello supone un impulso en la exportación de productos y conocimiento, algo esencial en el desarrollo de las relaciones entre los países. La traducción automática es una solución barata y rápida que permite el acceso a traducciones a todo el mundo, creando puentes de comunicación entre personas de todo el mundo.*

*Por otro lado, el aumento de datos, en el caso particular de los textos, permite la creación de conjuntos de datos que, dependiendo de la intencionalidad, partiendo de un único texto se pueden crear un número indeterminado de nuevas frases que, con el empleo de los modelos de lenguaje de hoy en día, consiguen mantener en gran mediad la calidad del texto inicial. Así, es posible crear textos más inclusivos, diversificados y que tengan en cuenta a una inmensa cantidad de personas, culturas y países.*

*Por último, el presente trabajo supone un gran impacto en la industria, en concreto en del lenguaje, e innovación, debido a que consigue mejorar la traducciones, tanto del ruso al inglés como del inglés al ruso, con el empleo de técnicas de aumento de datos, lo que posibilita un mayor entendimiento y una mejor comunicación. Además, el crecimiento económico también se ve influenciado, ya que al conseguir mejores traducciones, se consigue ahorrar en posibles post-procesos de corrección humana de dichas traducciones.*