**TRABAJO FIN DE GRADO**

# MECANISMO DE TRADUCCIÓN SEMI-SUPERVISADO DEL INGLÉS AL ESPAÑOL DE CONJUNTOS DE DATOS DE TEXTOS EMOCIONALES


CLARA LANDARÍBAR PAGAZAURTUNDUA

Grado en Ingeniería en Tecnologías Industriales

Especialidad de Ingeniería Electrónica y Automática

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

UNIVERSIDAD POLITÉCNICA DE MADRID


**2021**

# Agradecimientos

Estos años estudiando el grado han supuesto una etapa de aprendizaje y exploración como nunca antes había vivido. He tenido que enfrentarme a obstáculos dentro y fuera de la escuela, y he podido aprender de todos ellos una nueva lección. He conocido gente maravillosa, que ha tomado todo tipo de roles en mi vida: como guía, como compañía de risas, como pilar emocional y como todo ello al mismo tiempo.

He tenido la oportunidad de vivir fuera de mi casa y fuera de mi país, pero también la necesidad de volver al lugar donde crecí. He conocido el fracaso y la perseverancia, y me han acompañado a lo largo de este camino.

Con este trabajo, se cierra una etapa en la que me he podido desarrollar enormemente. Agradezco a todo el que ha jugado un papel, por pequeño que fuera.

Lo primero agradecer a mis tutores, Fernando y Mario, por acompañarme durante este proceso y por lo que me han enseñado.

Quiero agradecer a mis compañeros de clase por haberme acompañado a lo largo de estos años y por haberme dejado conocer de todos algo que me ha dejado huella. Gonzalo, Mara, Jaime, Marta, Loren, Dani, Pablo… podría nombrarles a todos aquí, pero creo que no es necesario.

A Belén. Gracias por ser mi compañera de vida paralela. Por mostrarme tanto. Por todo. Sin tí, todo este camino habría sido mucho más gris.

Gracias Fer y Pilar, creo que no saben todo lo que han hecho por mí durante estos años. Eternamente agradecida, os desearé siempre lo mejor.

Gracias Aba por ser mi aliado napolitano. Por mostrarme el camino en la UNINA. Por tu amistad.

También agradecer a las amistades de siempre y a las nuevas que he hecho en casa, en los meses en los que trabajé en este proyecto. Gracias Belén por hacerme sentir como una más.

Gracias Ben, por tus constantes palabras de apoyo, por cederme un espacio en tu casa y en tu vida. Gracias por creer en mí.

A toda mi familia, por estar presente.

A mis padres por creer en mí, por no dudar cuando yo lo hacía. Por celebrar mis éxitos.

A María y Karmele. Gracias por levantarme cuando lo necesité. Gracias por acompañarme siempre. Por el vínculo. Os quiero.

*¿Cómo podría amar una cosa que no soy capaz de recordar?*

*-Natalia Ginzburg*

# Abreviaturas

NLP. Natural Language Processing (Procesamiento del lenguaje natural)

MT. Machine Translation (Traducción automática)

AI. Artificial Intelligence (Inteligencia artificial)

ML. Machine Learning (Aprendizaje automático)

DL. Deep Learning (Aprendizaje profundo)

DD. Daily Dialog

ED. Empathetic Dialogues

SMT. Statistical Machine Translation (Traducción automática estadística)

NMT. Neural Machine Translation (Traducción automática neuronal)

ST. Sentence Transformers (Transformador de oraciones)

LR. Linear Regression (Regresión lineal)

NLU. Natural Language Understanding (Comprensión del lenguaje natural)

NLG. Natural Language Generation (Generación del lenguaje natural)

# Resumen

Este proyecto pertenece a la línea de investigación del robot asistente social POTATO del Grupo de Control Inteligente (GCI) del Centro de Automática y Robótica, y está enmarcado dentro del campo del Procesamiento del Lenguaje Natural (NLP), la rama de la Inteligencia Artificial que se encarga de las interacciones del lenguaje humano con las computadoras. Es un campo que está experimentando desde hace años un enorme crecimiento a nivel global. Dentro del Procesamiento del Lenguaje Natural existen diversas aplicaciones, pero las que se van a emplear en este proyecto son la Traducción Automática y las Métricas de Evaluación.

Las bases de datos disponibles hoy en día para el entrenamiento de agentes conversacionales son, en su gran mayoría, en inglés debido a que se trata de la lengua vehicular en el ámbito científico y de investigación.

Para el entrenamiento del agente conversacional POTATO, se desea emplear una base de datos en español que contenga una alta carga emocional, para tratar de mejorar la naturalidad con la que se desenvuelve en situaciones de diversa índole. Existen pocas bases de datos que cumplan con el requisito de estar basadas en emociones y las pocas que se pueden encontrar, son en inglés. Es por ello, que ante la inexistencia de bases de datos de calidad en español, se optó por realizar una traducción de dos bases de datos, DailyDialog y Empathetic Dialogues. Estas bases de datos muestran buenos resultados a la hora de mejorar la respuesta empática de los sistemas conversacionales.

Para realizar esta traducción, se empleó la herramienta de Microsoft Azure Translator, un servicio de Traducción Automática Neuronal basada en la nube a la que se accede mediante llamadas a una API REST, una interfaz que permite la interacción entre dos plataformas mediante un software. Una API tipo REST obtiene la información en formato HTTP y devuelve generalmente una respuesta en formato JSON. Las traducciones se realizaron y costearon por el GCI. Se obtuvieron de cada base de datos dos documentos de traducciones: uno con la traducción de la base de datos al español y otro con la traducción de vuelta al inglés de la base de datos traducida al español.

Las bases de datos están divididas en intervenciones, que son cada una de las unidades monológicas que suponen un cambio de voz. Estas intervenciones están codificadas y se agrupan en conversaciones. Se denominan de la siguiente manera: las intervenciones originarias en inglés, intervenciones originales; las intervenciones traducidas al español, intervenciones traducidas; y las intervenciones traducidas al inglés, intervenciones candidatas.

Las métricas de evaluación, se emplean para medir el rendimiento de un modelo entrenado en una tarea particular. Dentro de las posibilidades para la evaluación de traducciones automáticas, se ha decidido emplear una métrica tradicional de similitud léxica, BLEU y una métrica avanzada basada en embeddings por modelos entrenados con redes neuronales profundas.

BLEU es la métrica de referencia en el campo del NLP, basada en la superposción de n-gramas. La evaluación con BLEU de las bases de datos se emplea en el proyecto como preanálisis para obtener una valoración general, comparable con otros estudios por ser la métrica de referencia.

La evaluación avanzada de las bases de datos se ha realizado mediante la valoración de la similitud semántica textual. Para ello, se han empleado los modelos de Sentence Transformers. Estos modelos están basados en BERT (Bidirectional Encoder Representations from Transformers), que es un modelo preentrenado de conocimiento amplio, es decir, entrenado con corpus de texto sin temática específica. BERT es profundamente bidireccional (se tienen en cuenta tanto las palabra a la derecha como las palabras a la izquierda a la hora de contextualizar una palabra), lo cual permite obtener embeddings (representación numérica en forma de vector de una oración o de un conjunto de oraciones) con una mayor riqueza contextual. Para obtener una valoración de la similitud semántica textual, se hace uso de la similitud coseno, comparándose los embedding de la oración original y de la oración traducida, dando como resultado un valor entre cero y uno.

Sentence Transformers cuenta con multitud de modelos, entre los que se escogió para el análisis de las bases de datos el modelo XLM-R. Esta elección se realizó comparando la semejanza de los embedding multilingües disponibles en Sentence Transformers con un vector de puntuaciones que se evaluaron manualmente. XLM-R es un modelo multilingüe que obtiene un rendimiento muy alto en la comprensión linguística cruzada. En este caso, se emplearon las intervenciones originales y las intervenciones traducidas, ya que el modelo mapea los embeddings de diferentes lenguas al mismo espacio vectorial y son por tanto comparables entre ellos directamente.

Se evaluaron manualmente grupos aleatorios de intervenciones para encontrar errores recurrentes y revisar las evaluaciones tanto de la métrica BLEU como la del modelo XLM-R. Los errores son en general fácilmente modificables con un procesado de las bases de datos.

Además de evaluar las bases de datos con los dos métodos descritos, se entrenó un modelo de regresión lineal. Esta arquitectura buscaba relacionar la métrica de referencia BLEU con los modelos multilingües de Sentence Transformers. Se escogió como salida de entrenamiento el modelo monolingüe que obtuvo resultados más semejantes a los obtenidos por BLEU. El modelo se entrenó tanto con 2 entradas como con 6 entradas. Las entradas del modelo fueron los resultados del nivel de paráfrasis obtenida con los distintos modelos multilingües de Sentence Transformers. Se trata de una arquitectura novedosa que no se ha empleado en el campo de la evaluación con anterioridad.

**PALABRAS CLAVE**: métricas de evaluación, BLEU, Sentence Transformers, traducción automática, regresión lineal, inteligencia artificial, base de datos, modelo multilingüe, modelo monolingüe.

# Índice general

# MEMORIA

# Capítulo 0. Introducción

## 0.1 Motivación del proyecto

El trabajo pertenece a la línea de investigación del robot asistente social POTATO, desarrollado por el Grupo de Control Inteligente (GCI) del Centro de Automática y Robótica UPM-CSIC (CAR). El trabajo se enfoca en la creación de un conjunto de datos de entrenamiento para Sistemas de Diálogo emocionales en español para el robot de asistencia personal POTATO, mediante el uso de técnicas del campo del Procesamiento del Lenguaje Natural (NLP) e Inteligencia Artificial (AI). Con estos datos se podrá mejorar la capacidad de diálogo emocional en español que serán capaces de expresar los robots en el futuro.

El sector de la AI es sin duda uno de los que está experimentando mayor crecimiento a nivel global y dentro de este, el NLP es uno de los campos de investigación más potentes por su uso, cada vez mayor, a la hora cubrir distintas necesidades del mercado. Mordor intelligence[1] estima que el mercado global del NLP alcanzará un valor de 48.46 mil millones de dólares en 2026, siendo su valor actual de 10.72 mil millones de dólares.

Dentro de este marco, destaca el foco continuado en el inglés, por ser la lengua vehicular del marco científico y de investigación. A pesar de ello, en el NLP está cobrando cada vez más fuerza su variante multilingüe. En 2019 se introdujo el innovador modelo multilingüe XLM-R, el primer modelo multilingüe en obtener mejores resultados que los tradicionales modelos monolingües. Compañías como Google o Facebook están invirtiendo para el desarrollo y mejora de este tipo de modelos multilingües. Por ello, es especialmente interesante crear una base de datos de conversaciones en español, ya que, por el momento, existe un número muy reducido de las mismas. Como estudiante de ingeniería, resulta realmente motivador estudiar este tipo de tecnología que está en pleno desarrollo.

## 0.2 Objetivos

Los objetivos que se tratarán de cumplir durante el desarrollo del proyecto son los siguientes:

- Aprendizaje de conceptos y algoritmos básicos de aprendizaje automático (ML).
- Aprendizaje de conceptos y algoritmos básicos de la traducción automática (MT).
- Estudio del estado del arte mediante publicaciones relacionadas.
- Colaboración con el Grupo de Control Inteligente en líneas de investigación.

[1] https://www.mordorintelligence.com/industry-reports/natural-language-processing-market

- Análisis de la calidad de traducciones mediante mecanismos de reconocimiento paralelo, el uso de etiquetas y herramientas de traducción.

Además, durante el desarrollo del trabajo, se han ido trabajando nuevas habilidades y conocimientos según han sido necesarios. Algunas de estas son:

- Aprendizaje del lenguaje de programación Python y sus diferentes librerías como Numpy Pandas y otras que se mencionan a lo largo del documento.

- Aprendizaje del uso del entorno Colab de Google.

## 0.3 Materiales empleados

Como hardware, se ha empleado un HP Pavilion x360 Convertible con procesador Intel i7 CPU de 2,70 GHz y 12 GB de memoria RAM. El sistema operativo empleado fue Windows 10 Home 64.

## 0.4 Estructura del documento

El documento está compuesto de las siguientes partes:

- Memoria

- Organización del proyecto

- Anexos

- Bibliografía

La Memoria es la parte principal de este documento. Se compone de seis capítulos:

- Capítulo 0. Se da una introducción general del tema a tratar en el trabajo, los objetivos que se pretenden alcanzar y se añaden los materiales empleados para la realización del proyecto.

- Capítulo 1. Se trata del estado del arte sobre el procesamiento del lenguaje natural, la lingüística, la traducción automática y las métricas de evaluación automáticas.

- Capítulo 2. Se expone el marco teórico del proyecto. Se explican en este capítulo las herramientas empleadas en las diferentes etapas del flujo de las bases de datos empleadas.

- Capítulo 3. Se explica el desarrollo del proyecto, el proceso de obtención de los resultados.

- Capítulo 4. Contiene un análisis de los resultados obtenidos y se realiza una discusión de estos.

- Capítulo 5. Se explican las conclusiones y las posibles líneas futuras que podría tener el proyecto.

La organización del proyecto contiene un análisis de la gestión del proyecto, con una planificación temporal que incluye un diagrama de Gantt, y un presupuesto estimado del proyecto.

En los anexos, se incluye un índice de figuras y un índice de tablas.

Por último, la bibliografía muestra todas las referencias empleadas para realizar este proyecto.

# Capítulo 1. Estado del arte

## 1.1 Procesamiento del Lenguaje Natural (NLP)

### 1.1.1 Introducción

Una lengua natural es aquella forma de lenguaje humano con fines comunicativos que está dotado de una sintaxis y se ha generado de manera natural. Según Charles F. Hockett [1] tiene las siguientes características:

1. Canal vocal-auditivo
2. Transmisión irradiada y recepción direccional
3. Transitoriedad
4. Intercambiabilidad
5. Retroalimentación
6. Especialización
7. Semanticidad
8. Arbitrariedad
9. Carácter discreto
10. Desplazamiento
11. Productividad
12. Transmisión tradicional
13. Dualidad o doble articulación

Es un campo a medio camino entre la inteligencia artificial y la lingüística que busca diseñar programas o sistemas informáticos capaces de emular la conducta lingüística humana y de utilizar lenguajes naturales. Se creó por un lado para satisfacer el deseo de comunicarse con los ordenadores y por otro, para facilitar el trabajo de los usuarios que se manejaban solamente con lenguajes naturales y desconocían lenguajes de tipo máquina.

El procesamiento del lenguaje natural o como se dice en inglés Natural Language Processing (NLP) tiene hoy en día una creciente importancia por el uso que tiene en campos como el Machine Learning, la detección de spam en el correo electrónico, la extracción de información y muchas otras [2].

Aunque la mayoría de las personas que trabajan en el NLP son programadores y gente relacionada con las ciencias de la computación, también lingüistas, psicólogos o filósofos han mostrado interés en este campo.

### 1.1.2 Historia de la MT

#### 1.1.2.1 Precursores

La primera vez que se sugirió el uso de elementos mecánicos para superar las barreras del lenguaje se remonta al siglo XVII [3]. El decreciente uso del latín como lengua vehicular y el creciente uso de la comunicación científica dieron como resultado un

enorme deseo por encontrar una lengua internacional que supliera esas dificultades de comunicación y fuera tanto lógica como racional para poder emplearse en la comunicación de tipo científica.

Descartes en 1629 escribía: “Mettant en son dictionnaire un seul chiffre qui se rapporte à aymer, amare, philein et tous les synonymes [d’aimer dans toutes les langues] le livre qui sera écrit avec ces caractères [les numéros du code] pourra être interprété par tous ceux qui auront ce dictionnaire”[2]. Diccionarios de este tipo llegaron a ser publicados como es el caso de Cave Beck en 1657 [4]. Este escribió *“The universal Character*” que describía las reglas para una lengua universal que pudiera ser comprendida y empleada por cualquiera. Se basaba en una lista de 4000 ‘radicales’ a los cuales asignaba un valor numérico en orden alfabético (del 1 al 3996) a los que añadía prefijos y sufijos en forma de letras (e2518 = labour, p2518 = labourer) para formar los diferentes términos de la lengua. Existen otros casos como el de Athanasius Kircher en 1663 y Johann Joachim Becher en 1661 que también escribieron libros con propuestas de lenguas universales. A pesar de que ninguna de estas propuestas no pudiera superar de manera satisfactoria las dificultades que elevan las diferencias semánticas, son consideradas de alguna manera como precursores de la traducción automática.

Las sugerencias de este tipo de diccionarios de base numérica continuaron apareciendo durante los próximos siglos con ejemplos conocidos como el esperanto, pero no fue hasta 1933 que comenzaron a sentarse las bases de lo que hoy conocemos como traducción automática y procesamiento del lenguaje natural. Hasta ese momento, las herramientas ‘mecánicas’ que se habían propuesto requerían de una persona que las empleara. Sin embargo, en 1933, se emitieron dos patentes (Francia y Rusia) con diccionarios mecánicos. El francés Georges Artsrouni patentó “Mechanical brain”, un dispositivo mecánico que funcionaba con un motor eléctrico para registrar en una banda de papel la información. Cada línea contenía la palabra de entrada y equivalentes en otras lenguas. Para cada entrada había, en una segunda banda, perforaciones que codificaban a modo de selector.

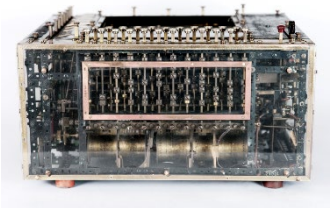


**ILUSTRACIÓN 1. “Mecanichal brain” patentada por Georges Artsrouni.[3]**

En Rusia, Petr Petrovich Smirnov-Troyanskii patentó una “máquina para la selección e impresión de palabras mientras se traducían de una lengua a otra o a otras simultáneamente.”. En este caso, se consideraban tres etapas para el proceso de traducción. En la primera etapa, una persona que conociera solamente la lengua de origen debía analizar el texto de una cierta manera lógica, extrayendo la base de cada palabra y asignando funciones sintácticas a cada oración. En la segunda etapa la máquina transformaba las secuencias en símbolos lógicos de la lengua origen en

[2] “Poner en el diccionario una sola cifra para aymer, amare, philein y todos sus sinónimos [de la palabra amor en todas las lenguas] resultará en que el libro escrito por medio de estas cifras [códigos numéricos] será legible para todos los usuarios de este diccionario.” [48].
[3] https://cafetran.freshdesk.com/support/discussions/topics/6000050879

símbolos lógicos de la lengua destino. En la última etapa, una persona que conocía la lengua destino convertía esta secuencia en la forma normal de su lengua. Se trataba de alguna manera de un diccionario mecánico, pero Troyanskii pensaba que el proceso de análisis lógico podía ser mecanizado por máquinas contraídas para ese propósito. Troyanskii mejoró su máquina e incluso tenía un proyecto de máquina electromecánica que no llegó a materializarse porque no tuvo apoyo por parte del gobierno ruso.

#### 1.1.2.2 Los orígenes

Se puede decir que fue en los años 40 a raíz de la Segunda Guerra Mundial cuando se incrementó el interés por estas disciplinas y por tanto se desarrolló la MT y más adelante el NLP, que engloba la MT y otros estudios de la lingüística y la AI. Se crearon máquinas como la ENIAC para calcular las tablas de disparo de proyectiles o la Colossus para descifrar mensajes militares alemanes. Tras la guerra, se establecieron numerosos centros de máquinas de cálculo en varios países. Aunque las primeras aplicaciones estaban más relacionadas con la física y las matemáticas, pronto comenzaron los primeros estudios sobre aplicaciones no numéricas.

Una de las aplicaciones que más destacó fue la de Alan Turing, considerado uno de los pioneros de lo que hoy conocemos como AI. Una máquina de Turing [5] es un modelo computacional que realiza una lectura/escritura de manera automática sobre una entrada llamada cinta con unas determinadas condiciones iniciales (configuración-m), generando una salida en esta misma que es de tipo numérico (0 o 1). En 1947 Turing [6] mencionaba distintas maneras en las que las máquinas demostrarían inteligencia: Estimación de errores; Lógica simbólica; Filosofía matemática; Juegos como el ajedrez; Traducción de diversos idiomas. En 1946 en Estados Unidos, A.D.Booth, consiguió que Warren Weaver, criptólogo y vicepresidente de la Fundación Rockefeller apoyara el estudio a gran escala de investigaciones a sobre MT [7] lo que fue determinante para darle importancia al estudio de la MT.

#### 1.1.2.3 Años 40-50. Altas expectativas

A partir de este momento y en los siguientes 10 años hubo un buen periodo para la lingüística computacional. En 1948 Richard H. Richens da las primeras soluciones al problema del análisis morfológico automático. Richens propuso la separación de las palabras en raíces (“stems”) y finales (“endings”) para reducir el tamaño de los diccionarios y además introducirles información gramatical. Escribió junto a Booth estos resultados en forma de memorándum [8], el cual mejoraron y presentaron en una conferencia de la MIT[4] en 1952. Warren escribió un memorándum en 1949 [9] y se lo compartió a 200 conocidos que consideró podían tener interés en conocer el concepto de MT. Para prácticamente la totalidad de los destinatarios se trataba de la primera vez que oían hablar de la MT y resultó fundamental, ya que surgió una rama empresarial relacionada con la traducción automática a raíz de ello. En este memorándum, Weaver se centraba en estrategias generales y hablaba de algunos de los principales problemas que podían darse en la comunicación en tres niveles: técnicos, semánticos y de influencia. Dentro de estos niveles escribió sobre: los problemas de los significados múltiples, la base lógica del lenguaje, la aplicación de la teoría de comunicación y

[4] MIT. Massachusetts Institute of Technology.

técnicas de criptografía y sobre las posibilidades de las invariantes universales del lenguaje.

El memorándum de Weaver tuvo mucha repercusión y se crearon líneas de investigación en el MIT, UCLA[5] y RAND[6]. Para solucionar la cuestión de los múltiples significados en las traducciones palabra por palabra, se introdujeron conceptos relacionados con el contexto y con los tipos de palabra que componían la oración. También comenzaron a cuestionar la traducción palabra por palabra en determinadas lenguas como el alemán y a proponer opciones como una codificación gramatical de las palabras indicando funciones y casos que no siempre se llegaron a poner en práctica.

En 1951, Bar-Hillel [10] hizo una encuesta sobre la situación de la MT en aquel momento. Yehoshua Bar-Hillel fue la primera persona que se fijó para trabajar únicamente en la investigación de la MT y desde esta posición pudo plantear temas que dominaron en los siguientes años como: la fiabilidad de la MT automática, el papel del post y pre procesado, los objetivos del análisis sintáctico, el papel de la información estadística, la posibilidad de una gramática universal, las fundaciones lógicas del lenguaje o el uso de vocabularios restringidos.

Bar-Hillel organizó en 1952 la primera conferencia acerca de la MT en el MIT con el patrocinio de la Rockefeller Foundation. Los asistentes venían de ámbitos diversos como ingeniería electrónica, lingüística, ciencias de la información o militares. En esta conferencia, se compartieron los avances que se estaban dando en todo el país y se propusieron ideas sobre nuevas líneas de investigación. La conferencia fue un éxito y los participantes estaban motivados a continuar sus investigaciones en esta área. A partir de este momento comienza a verse el término MT en artículos para el público general e incluso mencionado en libros de texto. Sin embargo, el resultado más relevante de la conferencia de 1952 fue la fundación de un equipo de investigación sobre MT en la Georgetown University encabezado por Leon Dostert, uno de los asistentes de la conferencia, para demostrar la viabilidad práctica de la traducción automática mediante un experimento piloto. Dostert era totalmente consciente de los considerables problemas lingüísticos de la MT, pero concluyó que sería más fructífero realizar un experimento de menor alcance, pero con implicaciones amplias que intentar resolver teóricamente una gran parte del problema. Dostert por tanto tomó la vía empírica y se alió con IBM[7] para llevar a cabo este experimento Georgetown-IBM. El programa estaba listo a finales de 1953 y se hizo una demostración pública de su funcionamiento en 1954 [11] en el Departamento de Computación Técnica de IBM en Nueva York. Esta traducción a pequeña escala del ruso al inglés fue uno de los acontecimientos más importantes de la historia de la MT. No obstante, hay que tener en cuenta que las condiciones en las que se realizó este experimento eran muy controladas debido a un vocabulario limitado, ejemplos escogidos con cuidado y pocas reglas gramaticales. En cualquier caso, mostraba que era un objetivo viable y ayudó a la inversión en investigación.

[5] UCLA. University of California, Los Angeles.
[6] RAND. RAND (Research and Development) Corporation.
[7] IBM. International Business Machines Corporation.

### 1.1.2.4 Años 60-70. Años invisibles

A finales de los años cincuenta, existían aproximadamente 20 grupos de investigación sobre el tema de la MT, con 500 investigadores alrededor de todo el mundo. Sin embargo, las expectativas que se habían creado se vieron pronto sacudidas por la realidad. Se observaron pocos resultados y con poca repercusión práctica. El informe de Bar-Hillel en 1960 [12], el que fuera cabeza de la investigación sobre MT en el MIT, junto con la opinión negativa de la National Academy of Science con su informe ALPAC en 1966 [13], resultaron en la retirada de los fondos de la mayor parte de los grupos de investigación. Por su parte el informe de Bar-Hillel concluía que la traducción totalmente automática de alta calidad no era un objetivo razonable. Argumentaba que no existía manera de escribir programas con la profundidad de conocimiento lingüístico que poseía un traductor en ese momento. También exponía que el tamaño insuficiente de las memorias o el aferramiento a un objetivo inalcanzable eran razones por las que la MT no se había desarrollado suficientemente. Añadía que el uso de maquinaria electrónica podía ser práctico en unos años, cuando las técnicas hubieran mejorado. El interés por la MT se redujo considerablemente en los siguientes años, aunque caben destacar proyectos como LOGOS (traducción de manuales de exploración armamentística), SYSTRAN (se empleaba en la NASA y en EURATOM), CULT (traductor en la Universidad China), ALP (traductor de textos mormones) y METAL (sistema de traducción alemán-español).

### 1.1.2.5 Años 70-80. Renovado interés

A partir de 1975, y más en los años 80, se vuelve a observar un crecimiento en el interés por la MT en Japón, Canadá y Europa. Existían en 1982 doce grupos operativos en Japón y once entre EEUU y Europa. Aumenta considerablemente el volumen de documentos traducidos (correcta o incorrectamente). El enfoque de investigación predominante en estos años se caracterizó por una adopción de la aproximación basada en la transferencia, orientada a la sintaxis y fundada en la formalización de normas gramaticales y léxicas influidas fuertemente por las teorías lingüísticas del momento. Se buscaba solventar problemas bien definidos y de esa manera se consiguieron mayores avances. Sistemas como LOGOS, SYSTRAN o METAL estaban diseñados para una aplicación general, pero en la práctica tenían unos diccionarios aplicados a usos específicos. Esto era una práctica común en los años 70 y 80 cuando, además, comenzaban a encargarse sistemas de traducción para empresas como Ford, que tenían un estricto control de vocabulario y sintaxis para una mínima revisión posterior.

A finales de los 80 [14], comienza a cobrar más importancia un nuevo enfoque basado en los corpus (uso de corpus bilingües, métodos estadísticos y aproximaciones basadas en ejemplos). La principal diferencia con el enfoque que se empleaba anteriormente es que no se usan reglas semánticas ni sintácticas en el análisis de textos o la selección de equivalentes léxicos. Los investigadores se sorprendieron del nivel de corrección de los textos traducidos empleando métodos estadísticos. Además, se beneficiaron de las mejoras en la velocidad de cálculo de los ordenadores y de las grandes bases de datos disponibles. Destacan las investigaciones del IBM basadas en métodos estadísticos.

#### 1.1.2.6 Años 90. Avances relevantes

Gracias a la disponibilidad de enormes cantidades de datos comienzan a desarrollarse otros métodos como computación paralela, redes neuronales y conexionismo. Estos modelos se entrenan para reconocer las relaciones más fuertes entre categorías gramaticales y entre elementos léxicos. También se continúa con la investigación y experimentación de sistemas basados en normas como METAL o el proyecto LMT. Surgen, por un lado, la tendencia lexicalista (la tendencia a cambiar las explicaciones sobre construcciones por explicaciones sobre palabras) y por otro, los formalismos basados en restricciones. Se comienzan a desarrollar también modelos híbridos que combinan lo mejor de los métodos basados en normas, estadísticos y en ejemplos.

Con la llegada de internet se aceleró mucho el desarrollo de la MT. Comenzaron a demandarse traductores de páginas web y servicios de mensajería online.

#### 1.1.2.7 NLP: Crece el término

A lo largo del documento se hace hincapié en la historia de la MT, por una parte, por su importancia dentro del NLP, porque fue la primera aplicación que se desarrolló dentro de este campo y está más íntimamente relacionada con este trabajo, pero cabe destacar que se desarrollaron paralelamente otras investigaciones en diferentes campos de la lingüística computacional. Este conjunto de aplicaciones se engloba en el término NLP. A mediados de los años 50, la publicación de Chomsky sobre las estructuras sintácticas introdujo el concepto de la gramática generativa que mejoró la ayuda que los lingüistas podían ofrecer a los investigadores de la MT. De esta manera, comenzaron a surgir otras ramas de investigación como el reconocimiento del habla.

En los últimos años, los mayores avances se han dado gracias a las enormes cantidades de datos accesibles para procesar y a tecnologías de DL. Así se han podido desarrollar aplicaciones como Google Translate, asistentes conversacionales como Siri o Alexa y para buscadores de internet.

### 1.1.3 Aplicaciones

Algunas de las aplicaciones del NLP son las siguientes:

- Chatbots (sistemas de diálogo).
- Correctores ortográficos y sintácticos.
- Clasificación de documentos.
- Generación automática de resúmenes.
- Traducción automática.
- Publicidad dirigida.
- Asistentes de voz (Siri, Alexa, etc.)
- Filtro de correos electrónicos.

### 1.1.4 Machine Learning

Se puede decir que el ML es una de las ramas de la AI pero lo cierto es que es una disciplina que abarca muchos más ámbitos, como estadística, psicología, teoría de la información y otras.

El objetivo principal del ML es diseñar y desarrollar algoritmos que permitan a los sistemas emplear datos empíricos, experiencia y entrenamiento para evolucionar y adaptarse a los cambios que ocurren en el ambiente.

Principalmente se pueden dividir los algoritmos de ML en dos clases principales:

- **Aprendizaje supervisado.** En el aprendizaje supervisado, se proporcionan ejemplos etiquetados de aprendizaje con información de entrada e información correcta de salida. De esta manera, se entrena el modelo con datos de entrada y salida controlados para obtener predicciones de salida con datos de entrada desconocidas por el modelo.

- **Aprendizaje no supervisado**. En el aprendizaje no supervisado, se proporciona únicamente la información de entrada y es el propio sistema el que depende de otras fuentes para determinar si está aprendiendo correctamente. Se trata de modelos que generalmente se emplean para categorizar conjuntos de datos donde es la red la que encuentra patrones que relacionan los datos.

Además, se considera una tercera clase, que es una combinación de las dos anteriores:

- **Aprendizaje semi supervisado.** Emplea una combinación de datos etiquetados y no etiquetados para el entrenamiento. Puede ser un útil cuando se trata de una gran base de datos

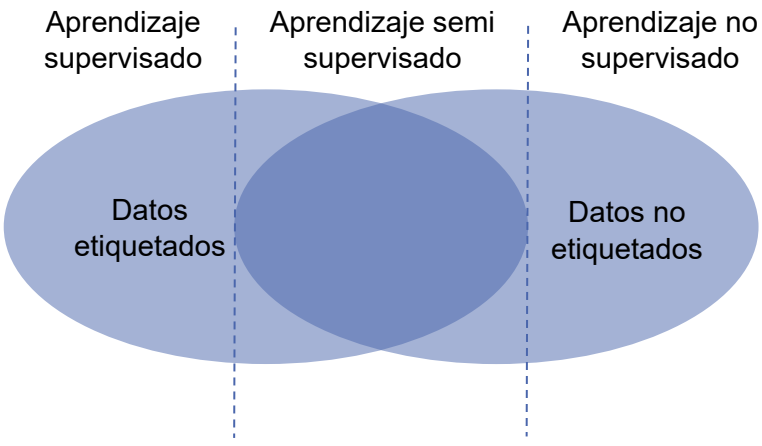



**ILUSTRACIÓN 2. Los algoritmos del ML divididos en categorías. [16]**

Dentro de estas clasificaciones más amplias, existen cantidad de algoritmos según las tareas y objetivos concretos que se desean obtener. En el siguiente gráfico se muestran algunos de los algoritmos más empleados en el NLP.

De manera general, los pasos que se deben seguir a la hora de emplear ML son los siguientes:

1. **Recolectar datos.** Se trata de un paso crucial. Los datos recolectados deben ser relevantes y una elección incorrecta de características o un número limitado de las mismas podría dar lugar a un modelo inefectivo.
2. **Preparar los datos.** Una vez recolectados los datos, el siguiente paso es prepararlos para su uso. Primero, se aleatoriza el orden de los datos para que no tenga peso en las elecciones del modelo. Además, se comprueba la oblicuidad de los datos de forma que sea equilibrada para evitar la parcialidad del modelo.
   Por último, se divide la base de datos en una parte mayor (aproximadamente el 80% de los datos) para el entrenamiento del modelo y una parte menor (alrededor del 20%) para la posterior evaluación.
   También se realizan refinamientos como la eliminación de duplicados o rechazo de lecturas incorrectas.
3. **Escoger un modelo.** Una vez se tienen los datos preparados, se escoge el modelo que más se adapta a los mismos y al objetivo que se desea alcanzar.
4. **Entrenamiento del modelo.** Es la parte esencial del ML. Se emplean datos que se han separado para el entrenamiento y se da un valor inicial a los pesos. El valor de estos pesos se va ajustando hasta minimizar el error en la salida de la predicción y el resultado esperado.

   - **Entrada.** Dato que entra en el modelo.
   - **Característica.** Cada una de las variables estudiadas.
   - **Peso.** Cantidad que cuantifica el peso de cada característica dentro del modelo. Este valor se modifica con el entrenamiento del modelo.
   - **Salida.** Valor que sale del modelo.
   - **Predicción.** Valor de salida obtenido por el modelo con una entrada tras el entrenamiento de este.

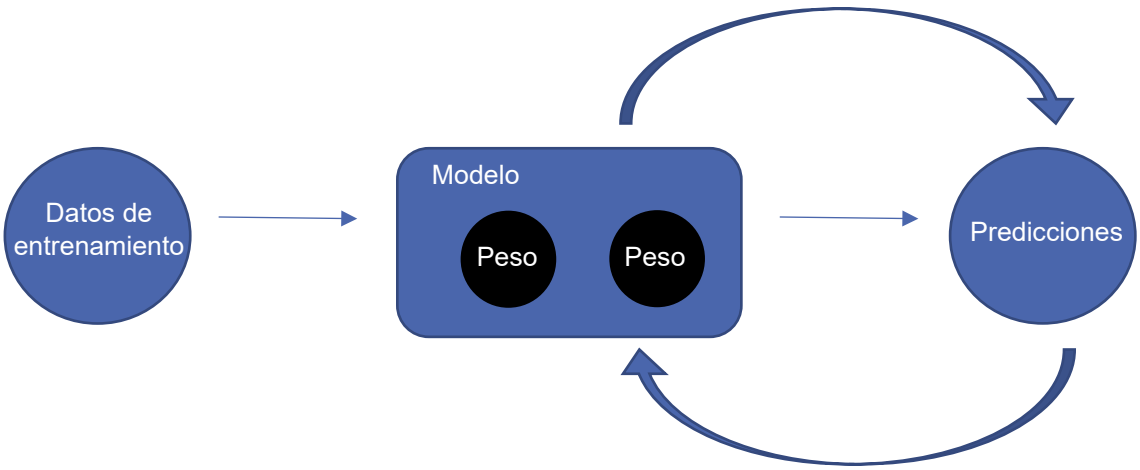



**ILUSTRACIÓN 3. Esquema del proceso de entrenamiento de un modelo.**

5. **Evaluación del modelo.** Una vez se ha entrenado el modelo, es necesario comprobar su correcto funcionamiento con datos reales. Para ello se emplean los datos reservados para la evaluación del modelo.
6. **Ajuste de hiperparámetros.** Si la evaluación es exitosa, se trata de encontrar la configuración de hiperparámetros que produce mejor rendimiento. Una de las

maneras, es realizando varios barridos de los datos de entrenamiento por el modelo aumentando la exposición a los mismos, pudiendo mejorar de esta forma la calidad del modelo. Otro modo es mediante el refinamiento de los valores iniciales de los pesos del modelo. El empleo de valores aleatorios puede dar lugar a malos resultados por lo que mejores valores iniciales o incluso una distribución en lugar de un valor pueden mejorar los resultados del modelo.

7. **Predicción.** El último paso del proceso es la predicción. Se considera el modelo listo para emplearlo en aplicaciones reales.

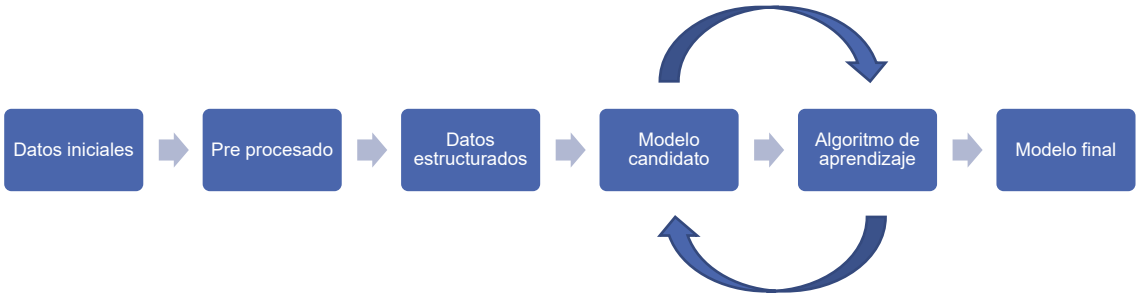



**ILUSTRACIÓN 4. Proceso completo de obtención de un modelo.**

Existen cantidad de modelos para la gran variedad de tareas que existen en el mercado. En la siguiente tabla, se muestran algunos de los más empleados dentro del ámbito del NLP.

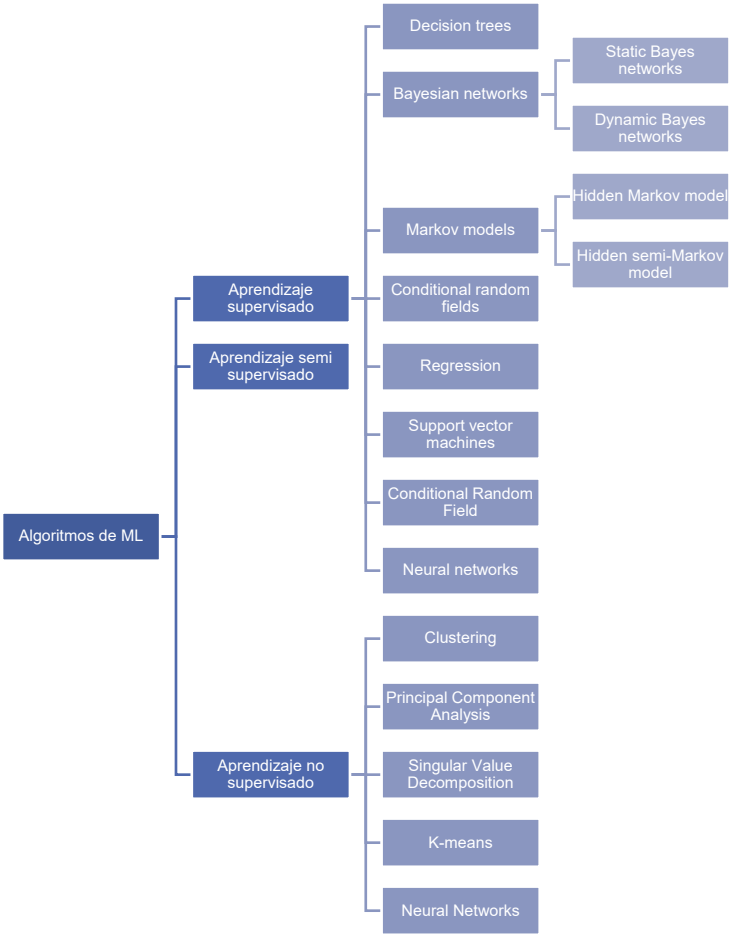



**ILUSTRACIÓN 5. Clasificación de los algoritmos de ML más empleados en NLP. [17]**

### 1.1.5 Deep Learning

El Aprendizaje Profundo (DL) es una rama del machine learning que tomó fuerza a raíz del crecimiento de enormes conjuntos de datos (lo que se suele considerar Big Data) que precisaban de avances en la tecnología convencional para su procesamiento.

Las estructuras del DL se suelen comparar con el sistema nervioso mamífero, ya que constan de una estructura que se asemeja de cierta manera, redes de neuronas (unidades de proceso) que son capaces de aprender de manera autónoma patrones dentro de los datos para realizar predicciones. Tiene infinidad de aplicaciones entre las que destacan el procesamiento de visión artificial y de lenguaje natural, detección de noticias fraudulentas o la conducción autónoma de vehículos.

Como se ha mencionado, la base del DL se encuentra en las Redes Neuronales, estructuras de múltiples capas de neuronas. Cada una de las neuronas es una unidad de procesamiento con entradas de información, con pesos y con una salida de información. Estas neuronas se organizan en sucesivas capas como se puede ver en la ilustración siguiente (Ilustración 3).

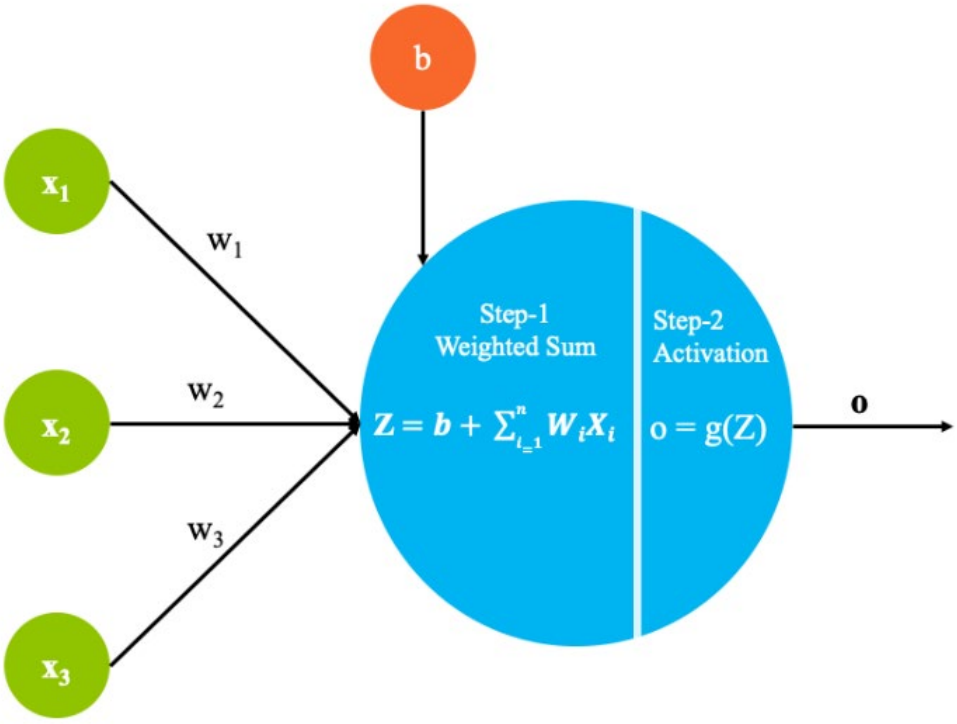



**ILUSTRACIÓN 6. REPRESENTACIÓN DE UNA NEURONA ARTIFICIAL.** [8]

Los pesos de cada neurona se van ajustando según la red va aprendiendo, confiriendo "más valor" a determinadas neuronas. Las salidas de las neuronas de una capa corresponden a las entradas de la siguiente y de esta manera, las redes son capaces de abstraer información relaciones y patrones entre los datos.

[8] https://developer.ibm.com/articles/an-introduction-to-deep-learning/

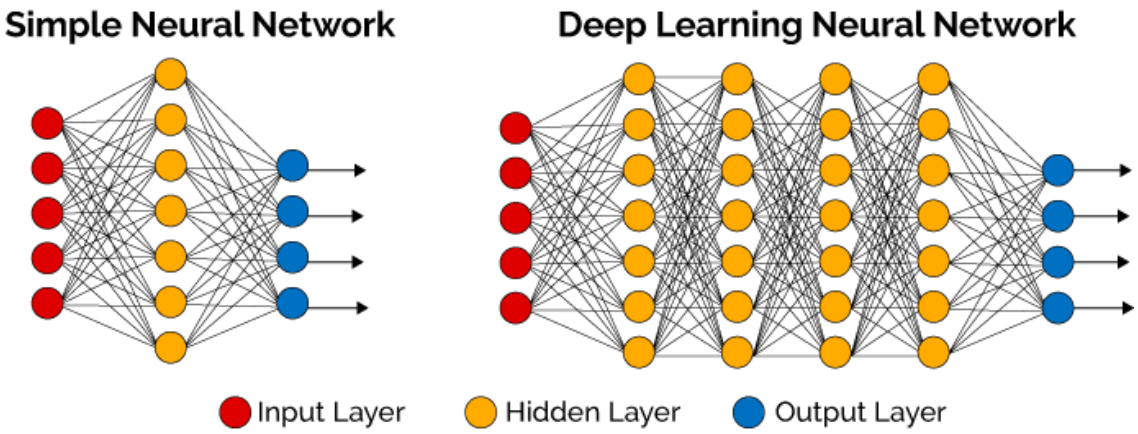



**ILUSTRACIÓN 7. Representación de una red neuronal simple y una profunda.**[9]

# 1.2 Lingüística

## 1.2.1 Fundamentos

La lingüística es la ciencia que se ha constituido en torno a los hechos de lengua [18]. Según Saussure, la tarea de la lingüística es:

a. Hacer la descripción y la historia de todas las lenguas que pueda ocuparse, lo cual equivale a hacer la historia de las familias de lenguas y a reconstruir en lo posible las lenguas madres en cada familia.
b. Buscar las fuerzas que intervengan de manera permanente y universal en todas las lenguas, y sacar las leyes generales a las que se puedan reducir todos los fenómenos particulares de la historia.
c. Deslindarse y definirse ella misma.

El lenguaje según Coseriu [19] es cualquier sistema de signos simbólicos (unidad resultado de la asociación de un significante y un significado [20]) empleados para la intercomunicación social. La lingüística en sentido estricto se encargaría solamente del estudio del lenguaje en que los signos son palabras constituidas por sonidos, es decir, del lenguaje articulado. Se describe lengua [18] como el conjunto de convenciones adoptadas por la sociedad para permitir el ejercicio del lenguaje. La lengua, al contrario que el lenguaje, es una totalidad en sí y un principio de clasificación.

Según M. Fernández, la lingüística se divide en diferentes áreas de estudio [20]:

- **Fonética**. Se ocupa de las proyecciones sustanciales de los significantes sonoros. Estudia los sonidos en su aspecto material, como realizaciones concretas.
- **Fonología**. Estudia los sonidos considerándolos significantes sonoros con un determinado valor lingüístico. Se encarga de la caracterización abstracta de sistemas de sonidos o signos.
- **Morfología**. Es la disciplina que se encarga del estudio de las palabras. Cómo se forman y su relación con otras palabras del mismo idioma.

[9] https://www.akademus.es/blog/tecnologia/que-es-el-deep-learning/

- **Sintaxis**. Se ocupa del estudio de la oración. Es el conjunto de reglas, principios y procesos que rigen la estructura de las oraciones en un lenguaje dado.
- **Semántica**. Estudia el significado, es decir, se ocupa del plano del contenido, de los significantes.
- Lexicología. Rama de la lingüística que se encarga de estudiar la estructura y el funcionamiento del repertorio léxico (vocabulario).
- Lexicografía. Disciplina que se encarga de la elaboración de diccionarios.

### 1.2.2 Lingüística computacional

La lingüística computacional según M. Villayandre [15] se enmarca según quién se ocupe de la definición en el campo de la lingüística (general, teórica o aplicada), en el de la informática (dentro de la AI o el NLP) o en un espacio de intersección entre ambas.

1. **Como parte de la lingüística**

   La LC (Lingüística Computacional) estudia los mecanismos que posibilitan la comunicación por medio del lenguaje con la ayuda de ordenadores. Las aplicaciones de la LC son más abundantes en la lingüística aplicada, ya que la finalidad es desarrollar aplicaciones concretas. La MT formaría parte de las aplicaciones de la lingüística aplicada.

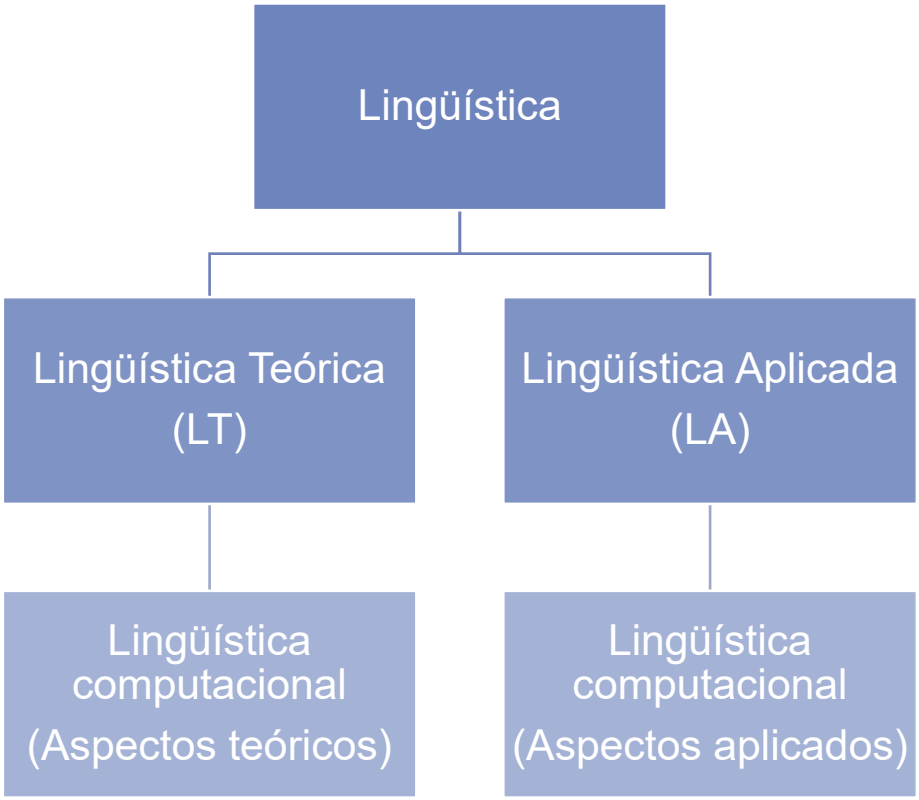



**Ilustración 8 La LC como rama de la LT y la [15]**

2. **Como parte de la informática**

   El lenguaje conecta la informática con la lingüística, pero se diferencia de esta en el objetivo final, que no es como en la lingüística una, mera indagación, sino que pretende reproducir una capacidad cognitiva en programas informáticos. Se puede enmarcar dentro de la AI, como una subdisciplina de esta. También se podría estudiar como una parte del NLP, incluso podría decirse que son términos casi equiparables en términos generales. En este trabajo se considera que la lingüística computacional es una subdisciplina del NLP.

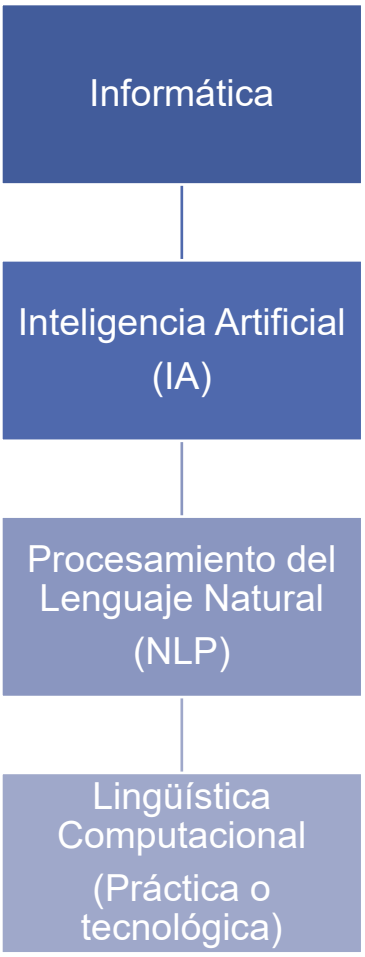



**Ilustración 9 La LC como NLP [15]**

## 1.2.3 Inglés y español: características lingüísticas

Siendo el inglés la primera lengua más hablada a nivel global y el español la tercera, es clara la importancia que tiene su conocimiento y su aplicación en entornos como el del presente trabajo, el NLP. Es conveniente comentar algunas de las características de cada una de estas lenguas.

#### 1.2.3.1 Español

El español[10] es una lengua indoeuropea de la subfamilia romance procedente que cuenta con 586 hablantes en todo el mundo. Tipológicamente es una lengua flexiva (incluye información mediante morfemas) fusionante (tendencia a fusionar morfemas, de tal forma que un morfema realiza simultáneamente el papel de varios morfemas teóricos[11]), de núcleo inicial (el núcleo sintáctico ocupa la posición inicial dentro del sintagma[12]) y el orden básico es SVO (sujeto verbo objeto).

#### 1.2.3.2 Inglés

El inglés[13] es una lengua indoeuropea del grupo germánico occidental. Debido a las invasiones vikingas y posterior invasión normanda a las islas británicas, ha recibido gran cantidad de préstamos de las lenguas germánicas septentrionales y del francés. Además, posee una gran cantidad de léxico que proviene de cultismos latinos. Su orden básico es SVO (sujeto verbo objeto)

### 1.2.3 Diferencias entre el español y el inglés

Aunque estas lenguas guardan ciertas similitudes: ambas son alfabéticas (y utilizan el alfabeto romano) y comparten aproximadamente un 35% de su vocabulario (a estas palabras se las denomina cognados), pertenecen a familias lingüísticas diferentes por lo conviene resaltar las diferencias existentes entre ellas de cara a la evaluación de traducciones.

Las principales diferencias lingüísticas entre el español y el inglés son las siguientes.[14]

1. Género de los sustantivos.

   Mientras que en inglés los sustantivos carecen de género y los acompaña el artículo "*the*", en español son considerados masculinos o femeninos y los acompañan los artículos "*el*" y "*la*" respectivamente.

2. Orden de adjetivos y sustantivos.

   En español, los adjetivos se colocan generalmente detrás del sustantivo, mientras que en inglés se colocan al contrario. Por ejemplo, la frase "*el perro marrón*" se traduce como "*the brown dog*". Además, en español no solo los artículos concuerdan con el sustantivo, sino que también lo hacen los adjetivos, tanto en género como en número. Por ejemplo, "*the brown dogs*" sería "los perros marrones"

[10]https://es.wikipedia.org/wiki/Idioma_espa%C3%B1ol#Pol%C3%A9mica_en_torno_a_%C2%ABespa%C3%B1ol%C2%BB_o_%C2%ABcastellano%C2%BB
[11] https://es.wikipedia.org/wiki/Lengua_fusionante
[12] https://es.wikipedia.org/wiki/Par%C3%A1metro_de_posici%C3%B3n_del_n%C3%BAcleo
[13] https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s#Clasificaci%C3%B3n
[14] https://www.fluentu.com/blog/spanish/differences-between-english-and-spanish/

3. La negación

En español, la negación es doble. Como se puede comprobar en "*No entiendo nada*", tanto la palabra "*no*" como "*nada*" son negativas. En inglés, la traducción literal sería "*I do not understand nothing*", pero sería incorrecta pues emplean una única negación. La traducción correcta sería "*I do not understand anything*".

4. Los posesivos

En inglés, la fórmula apóstrofo más "s" significa pertenencia. En español se emplea la preposición "de". Por ejemplo, *"Benedict's office*" se traduce como *"La oficina de Benedict"*.

5. Conjugación

En español existe un número mayor de posibilidades a la hora de conjugar un verbo. Mientras que en inglés el verbo *"know"* se puede conjugar como *"know"*, *"knows"*, *"knew"*, *"known"* y *"knnowing"*, en español tiene muchas más posibilidades como "*sé*", "*sabes*", "*sabe*", "*sabía*", "*sabíamos*", "*supe*", "*supimos*" …

6. Sujeto omitido

En español, al contar con una conjugación compleja de los verbos, el sujeto se puede saber sin nombrarlo explícitamente. Es por ello, que es a menudo omitido. En español se puede decir "*yo pienso*" o "*pienso*" siendo ambas correctas. Sin embargo, en inglés el sujeto debe ser siempre explícito y la traducción de la frase anterior sería "*I think*".

7. Preposiciones

En español hay 23 preposiciones mientras que en inglés hay unas 150. Por ese desajuste, muchas preposiciones en inglés se traducen por una sola en español. Por ejemplo:

- La tarta está en el frigorífico. → The cake is in the fridge.
- Hay una planta en el suelo. → There's a plant on the floor.
- Estoy en la boda. → I'm at the wedding.

8. La palabra "*it*"

Se emplea constantemente en inglés, "*It's cold outside*", "*What is it?"* son solo algunos ejemplos de casos en los que no se traduce en español. Se debe a la razón mencionada anteriormente: la conjugación del verbo indica implícitamente el sujeto. Por ello, "*it seems*" se traduce como "parece" o "*it is small*" como "es pequeño".

9. La puntuación y las mayúsculas

En su mayor parte, son similares en ambas lenguas. Algunas de las diferencias son:

- En español, se añade un signo de puntuación al comienzo de la oración en el caso de exclamación y pregunta. En inglés solo se emplea al final de la oración. Ej. "How nice!" → "Qué agradable!"
- Los días del año, los meses y los idiomas se escriben en mayúscula en inglés.

### 1.2.4 Ambigüedad del lenguaje: problemática

Uno de los problemas a los que se enfrenta el campo del NLP es la ambigüedad del lenguaje. La ambigüedad lingüística es la cualidad que hace que un texto esté abierto a múltiples interpretaciones. Esto hace especialmente difícil para la AI codificar con fiabilidad sin contexto.

Por un lado, existe la ambigüedad léxica, que ocurre con gran frecuencia porque las palabras o las expresiones pueden tener múltiples significados. Por ejemplo, en la oración “Se citaron en el banco donde se habían conocido” la palabra “banco” puede significar el mobiliario urbano donde uno se sienta o una oficina de una entidad financiera. Esto supone un problema que se ve incrementado cuando se quiere realizar una traducción. En el caso de la MT, una oración de este tipo sin contexto se traducirá sin ningún tipo de certeza en la calidad del resultado.

Por otro lado, existe la ambigüedad estructural, que aparece cuando el orden de las palabras cambia dentro de la oración, cambiando el significado de esta. Un ejemplo sería la oración “Vio un hombre en un barco con un catalejo.”, donde la oración tiene varios significados por su estructura.

Además de estos tipos de ambigüedad, encontramos en el lenguaje figurado un gran problema de interpretación. El lenguaje figurado comprende la metáfora, la ironía, las frases hechas o los juegos de palabras. Esto es especialmente complicado de gestionar por las diferentes vertientes del NLP. Además, hay lenguas que tienen una enorme cantidad de expresiones idiomáticas y frases hechas, como es el caso del inglés, lo cual dificulta los trabajos de desambiguación.

## 1.3 Traducción Automática

La traducción automática es una de las aplicaciones más empleadas de la AI hoy en día. Se emplean tanto aplicaciones como servicios online que son capaces de traducir grandes cantidades de texto entre cualesquiera de las lenguas que soportan. La traducción automática aún está en desarrollo, pero existen tecnologías muy complejas y punteras que se están empleando para realizar esta tarea como pueden ser el DL, big data, lingüística, computación en la nube y APIs.

### 1.3.1 Traducción automática estadística/neuronal

Ambos métodos de traducción necesitan grandes cantidades de datos traducidos por humanos para entrenar sus sistemas. Además, ninguno actúa como un diccionario bilingüe, traduciendo palabras basándose en potenciales traducciones, sino que traducen basándose en el contexto de la palabra en la oración. Sin embargo, predomina enormemente el uso de la traducción automática neuronal por su rendimiento, recibiendo valoraciones mucho más altas con distintas métricas de evaluación automáticas, especialmente en idiomas complejos morfológicamente. [21]

#### 1.3.1.1 Traducción automática estadística

La primera técnica de traducción automática que se desarrolló fue la estadística (STM), a mediados de los años 2000. Se trata de un sistema que no trata de implementar normas de traducción entre lenguas, sino que aprovecha las traducciones humanas existentes para tratar de aprender cómo funcionan las transformaciones entre lenguas estadísticamente. Se emplean corpus paralelos, conjuntos de texto con frases, palabras y expresiones traducidas en parejas de idiomas para detectar correspondencias entre lenguas y encontrar la mejor traducción para frases de entrada.

#### 1.3.1.2 Traducción automática neuronal

Las redes neuronales comienzan a emplearse en la traducción automática a mediados de los años 2010. Una de las diferencias de la traducción estadística neuronal (NMT) frente a la SMT es que se toma la oración completa en contexto, no solamente algunas palabras cada vez.

Se emplea un mecanismo de codificador-decodificador para realizar las traducciones. Cada palabra se codifica con un vector que puede tener más de quinientas componentes. Estas componentes representan información sobre la palabra como puede ser el género, el número, el tipo de palabra, etc. Estos vectores entran en el sistema y se codifican en un vector mayor que añade información acerca del contexto de la palabra en la oración. Estos vectores se pasan por las distintas capas neuronales para afinar sus valores. Una vez obtenidos los vectores finales, se hacen pasar por una capa de atención que decide qué palabra debe traducirse a continuación y abandona aquellas que no se necesitan en la traducción. Se realimenta esta capa hasta finalizar la traducción de la oración.

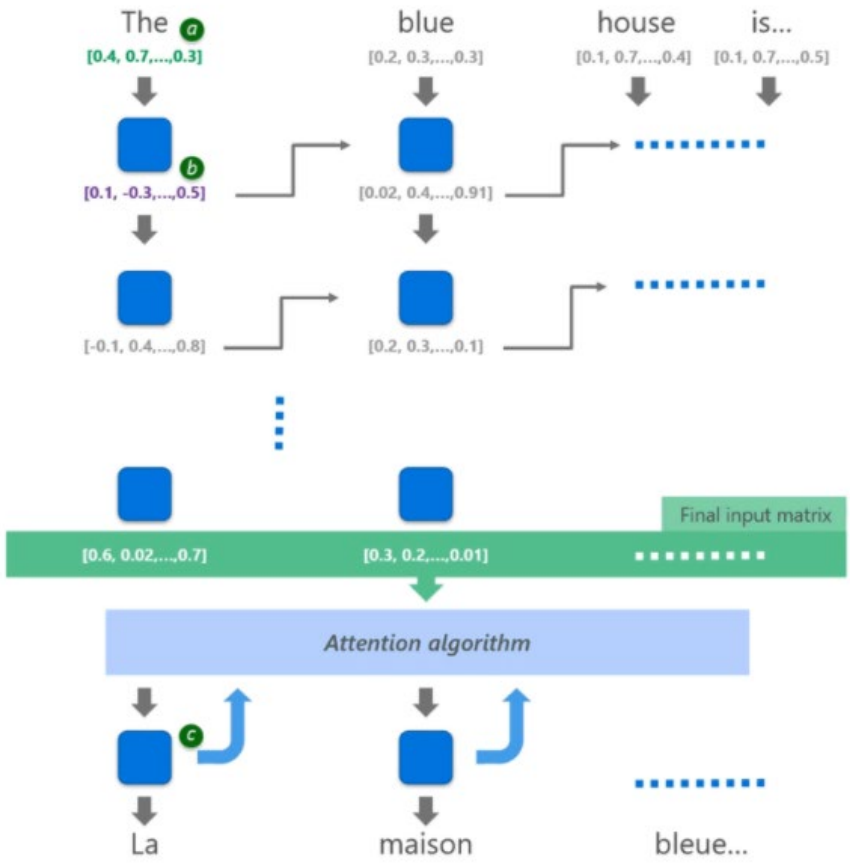



**ILUSTRACIÓN 10. Ejemplo de traducción neuronal.**[15]

## 1.3.2 Word Embedding y Sentence Embedding

El objetivo del *word embedding*, es la representación de palabras mediante vectores numéricos. Es uno de los conceptos fundamentales dentro del NLP y se emplea en multitud de contextos diversos. Además, con este principio del *embedding*, se han desarrollado modelos de última generación como BERT, del que se hablará en más adelante.

Como se ha mencionado, la idea principal del embedding es la representación de palabras mediante valores numéricos en forma de vectores. Cada uno de los valores, representa alguna de las características de la palabra como puede ser el género, el número o cualquier otra.

Como ejemplo se representa la palabra *king* (rey en inglés) de la siguiente manera:

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -
0.08813 , 0.47377 , -0.61798 , -0.31012 , -0.076666, 1.493 , -
```

[15] https://www.microsoft.com/en-us/translator/business/machine-translation/#nnt

```
0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 , -
0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 ,
0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927
, -0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -
0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585
, -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

Se trata de un vector de cincuenta componentes, del que es complicado decir algo viendo solamente los valores numéricos. En las siguientes ilustraciones, se puede observar la relación que diferentes palabras tienen entre ellas según su *embedding,* coloreado según el valor numérico de cada una de las componentes.

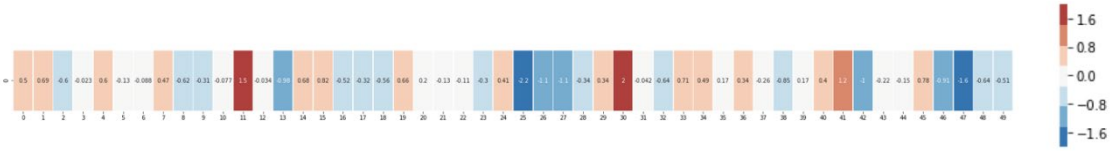



**ILUSTRACIÓN 11. Representación visual del embedding de la palabra en inglés *"King"*[16]**

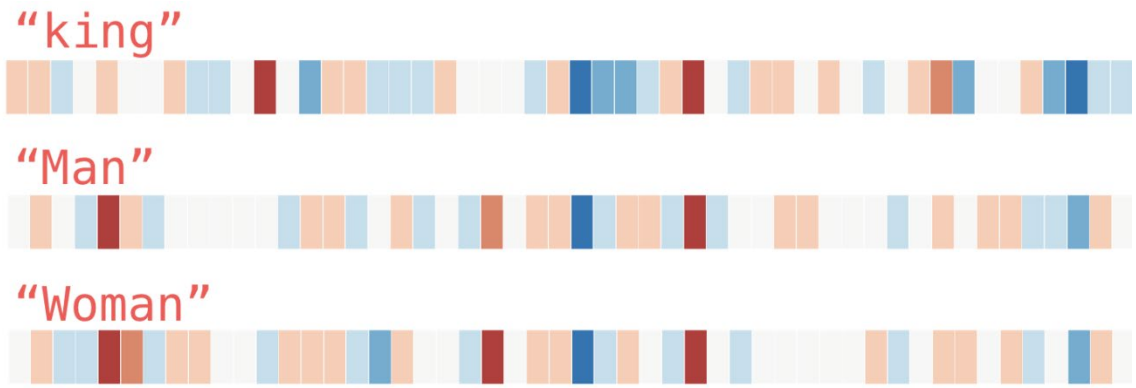



**ILUSTRACIÓN 12. Comparación de las representaciones de los embeddings de las palabras en inglés *"King"*, *"man"* y *"woman"* de manera visual.[17]**

De la misma manera, y también considerablemente utilizado se encuentra el *sentence embedding*. El principio es el mismo que el del *word embedding*, pero representando oraciones en lugar de palabras individuales mediante vectores. Ayuda a la comprensión del contexto y la intención a las diferentes herramientas del NLP. Uno de los usos del *sentence embedding* que se ha empleado en este proyecto, es el de la evaluación de similitud entre dos oraciones, mediante el uso de la similitud coseno.

[16] https://jalammar.github.io/illustrated-word2vec/
[17] https://jalammar.github.io/illustrated-word2vec/

## 1.4 Métricas de Evaluación de Traducción Automáticas

Una métrica de evaluación es un método para medir el rendimiento de un modelo entrenado, en este caso de un modelo de MT. Aunque existen métricas de evaluación manuales, tienen muchas desventajas ya que son lentas, caras y no reproducibles. Por estas razones, las métricas de evaluación manuales se han sustituido ampliamente por las métricas de evaluación automáticas como las que se describen a continuación. [22]

En el siguiente esquema se muestran algunos de los métodos de evaluación automática más empleados para evaluar modelos de MT.

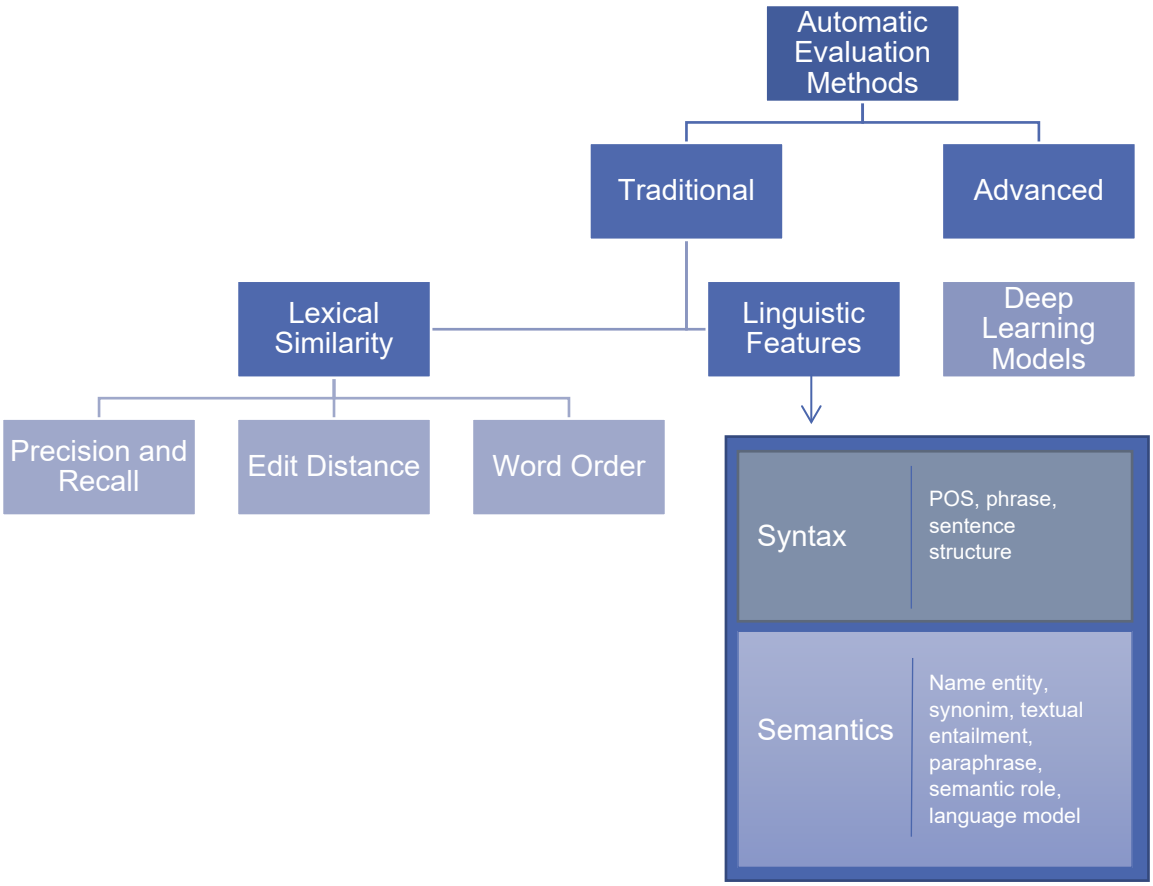



**ILUSTRACIÓN 13. Diagrama de árbol de los diferentes métodos de evaluación automática empleados en MT [22]**

La calidad de una traducción se suele medir según tres parámetros: *adecuación, fluidez y fidelidad*.

- Adecuación. Mide la cantidad de significado de la lengua origen que se ha transmitido en la lengua destino.
- Fluidez. Mide la calidad gramatical de las oraciones y su facilidad de interpretación.
- Fidelidad. Mide si la traducción reproduce con certeza el significado del texto de origen.

Para determinar la calidad de una traducción, se necesitan traducciones humanas que pueden ser únicas o múltiples. Estas traducciones se llaman traducciones de referencia y son aquellas que realizan profesionales en la materia. Pueden ser únicas, realizadas por un único traductor, lo cual puede ser limitante a la hora de evaluar, ya que existen diferentes posibilidades a la hora de escoger un término entre varios sinónimos y estructuras sintácticas, o pueden ser múltiples los traductores que elaboren las referencias.

Se emplean diferentes parámetros de medida como pueden ser la superposición de palabras, secuencias de palabras, orden de estas o distancias de edición. Estas últimas son las que se conocen como "métricas de similitud léxica". Métricas más desarrolladas tienen en cuenta además sintaxis y semántica, y se las denomina "características lingüísticas". Los casos más avanzados de métodos de evaluación automáticos hacen uso de DL.

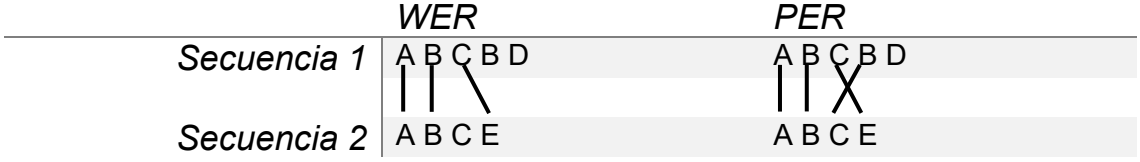
### 1.4.1 Métricas basadas en la similitud léxica

1. **Distancia de edición**

   Se calcula el número mínimo de pasos de edición para transformar la traducción a la referencia empleando la **WER** (Word Error Rate). Se tiene en cuenta el orden de las palabras y las operaciones incluyen sustitución (reemplazo de una palabra por otra), inserción (adición de una palabra) y supresión (eliminación de una palabra).

$$WER = \frac{sustitución + inserción + supresión}{referencia_{longitud}}$$

   El punto débil de WER es que el orden de las palabras no se tiene en cuenta de manera apropiada. Existen casos en los que una traducción "errónea" resulta ser válida. Por ello, existe una tasa de error que es independiente del orden, llamada **PER** (Position-independent Error Rate). PER cuenta el número de veces que se repite cada palabra de la traducción y compara con la referencia. Las palabras que no coinciden son inserciones o supresiones.




   También existe otra manera de evitar los problemas del orden de las palabras llamada **TER** (Translation Edit Rate): una métrica de error que calcula el número de ediciones requeridas para obtener una frase de referencia con el resultado de su traducción.

$$TER = \frac{n^{\underline{o}}\ ediciones}{n^{\underline{o}}\ palabras\ de\ referencia}$$

**2. Precisión y llamada**

La que se emplea de manera más generalizada es **BLEU (Bi-Lingual Evaluation Understudy)**. Es una métrica de evaluación basada en el grado de superposición de N-gramas (subsecuencia de n elementos de una secuencia dada) entre la oración traducida y la referencia de esta.
BLEU computa la precisión en un baremo de 1 a 4 con el coeficiente de penalización por brevedad (BP):

$$BLEU = BP \times exp \sum_{n=1}^{N} \lambda_n \log Precision_n$$

$$BP = \begin{cases} 1, & c > r \\ e^{1-\frac{r}{c}}, & c \leq r \end{cases}$$

Donde *c* es el total de palabras de la traducción, *r* el de la referencia y $\lambda_n$ el peso de precisión. Si hay múltiples referencias, se toma aquella que se aproxima más a la traducción candidata. Normalmente se selecciona $\lambda_n$ como un peso uniforme.

Para medir N-gramas más informativos existe la métrica **NIST** que tiene un peso de información añadido para palabras poco comunes.

$$Info = log_2(\frac{\#occurrence\ of\ w_1, \dots, w_{n-1}}{\#occurrence\ of\ w_1, \dots, w_n})$$

Además, sustituye la media geométrica de coocurrencias por una media aritmética de coincidencias de N-gramas.
**ROUGE** es una métrica orientada al recuerdo. Se emplea para sistemas de resúmenes automáticos.
Las medidas tipo F son aquellas que combinan precisión (*P*) y recuerdo (*R*) y se emplearon en tareas de recuperación de información y más tarde en extracción de información y otras tareas relacionadas con la MT.

$$F_\beta = (1 + \beta^2)\frac{PR}{R + \beta^2 P}$$

**METEOR** es una métrica basada en el concepto general de alineaciones de unigramas flexibles, precisión de unigramas y recuerdo de unigramas. Se tienen en consideración alineaciones de palabras con la misma raíz que son variantes morfológicas y sinónimos. Introduce un coeficiente de penalización que emplea el número de partes alineadas. METEOR no se emplea de manera generalizada y ha ido decreciendo desde la llegada de métodos DL.

$$Penalty = 0.5 \times \left(\frac{nº\ chunks}{nº\ matched\ unigrams}\right)^3$$

$$METEOR = \frac{10PR}{R + 9P} \times (1 - Penalty)$$

**3. Orden de palabras**

El orden correcto de las palabras juega un papel importante a la hora de asegurar una traducción de calidad. Sin embargo, la diversidad del lenguaje permite que oraciones con diferentes construcciones compartan significado y, por tanto, se estudia la penalización de aquellas oraciones que tengan un orden incorrecto y no de aquellas que tienen un orden "correctamente" diferente.
Basándose en la evaluación explícita del orden y la selección de palabras se desarrolló la métrica de evaluación **ATEC** [23] (Assessment of Text Essential Characteristics). La evaluación del orden se hace en términos de discordancia entre posiciones de palabras y secuencias de la oración candidata y su referencia. Para la evaluación de la selección de palabras se tienen en cuenta varios niveles lingüísticos como la raíz, el sonido o la forma. Además, se les dan pesos según la informatividad de cada palabra.
A raíz de esto, se han desarrollado otras métricas como:

- **PORT:** Combina información de la precisión, el orden y el recuerdo**.**
- **LEPOR:** Es una combinación de muchos factores de evaluación incluyendo penalizaciones por orden basadas en N-gramas y penalizaciones de precisión, recuerdo y longitud de oración. Esta métrica en particular ha dado muy buenos resultados en traducciones de inglés a otras lenguas.

## 1.4.2 Métricas basadas en características lingüísticas

Aunque algunas de las métricas mencionadas emplean información lingüística, los métodos de similitudes léxicas se centran en las coincidencias exactas de la traducción resultante.

**1. Similitud sintáctica**

**a. Part of speech (POS).** Se refiere en gramática a la categoría lingüística que ocupa una determinada palabra o artículo léxico (nombre, verbo, preposición, etc.). El modelo **IBM1**, permite evaluar la calidad de las traducciones calculando la similitud de puntuaciones entre la oración origen y su traducción empleando información de los morfemas, 4-gramas de POS y probabilidades de lexicón. La métrica de evaluación **TESLA** (Translation Evaluation of Sentences with Linear-programming-based Analysis), combina los sinónimos de tablas de frases bilingües e información de POS con coincidencia de N-gramas y relaciones de sinónimos de WordNet.

b. **Categorías sintagmáticas.** En lingüística, un sintagma se refiere a un grupo de palabras que funcionan como una unidad simple dentro de la oración. Para medir el rendimiento de un sistema de TA se exploran diferentes opciones como: el estudio de las construcciones sintácticas con un conocimiento lingüístico más complejo (identificación de proposiciones subordinadas adverbiales y sintagmas preposicionales, etc.); el uso del ratio de longitud, palabras desconocidas y número de sintagmas, asumiendo que estructuras gramaticales similares deberían ocurrir en origen y traducción; o el empleo del chunking, que consiste en la división de oraciones en subconstituyentes.

c. **Estructura de oraciones.** La sintaxis estudia los principios y procesos por los cuales se construyen las oraciones. Para medir la calidad de las oraciones se comparan las similitudes entre los árboles de dependencias. También emplean analizadores automáticos para identificar los constituyentes en las oraciones e ir enlazándolos a unidades de orden superiores (sintagmas, grupos nominales, etc.). La métrica RED emplea árboles de dependencia.

2. **Similitud semántica**

a. **Entidades nombradas.** Tomado de la tecnología NER que trata de identificar y clasificar elementos atómicos en el texto en diferentes categorías de entidad. Ejemplos de categorías de entidad son nombres de personas, localizaciones o tiempo.

b. **Sinónimos.** Son palabras con un significado cercano entre ellas. Se puede emplear una base de datos como **WordNet** que agrupa las palabras en conjuntos llamados synsets. Los synsets son estructuras jerárquicas con las palabras en diferentes niveles de acuerdo con sus relaciones semánticas.

c. **Vinculación textual.** Encuentra relaciones direccionales entre fragmentos de texto en sentido único. Si un párrafo (TB) hace que otro sea probablemente cierto (TA) existe una relación TB$\rightarrow$TA.

d. **Paráfrasis.** Es la reafirmación de un texto empleando palabras diferentes que podría verse como una vinculación textual bidireccional. Se emplea la métrica **TER-Plus** (**TERp**) que considera secuencias de palabras en la referencia como paráfrasis de secuencias en la hipótesis.

e. **Roles semánticos.** Se emplean como características lingüísticas por algunos investigadores. Un ejemplo de este tipo de métrica es **MEANT** que captura las relaciones predicado-argumento como la relación estructural en el ámbito semántico. Además, utiliza diferentes pesos según la importancia relativa a la preservación del significado.

f. **Modelos de lenguaje.** Asignan una probabilidad a una secuencia de palabras según una distribución de probabilidad.

g. **Árboles de permutación.** La métrica **BEER**, incorpora un gran número de características en un modelo lineal para maximizar la correlación con juicios humanos. Se exploraron dos tipos de rasgo: n-gramas de caracteres y árboles de permutación. En estos últimos, se investiga más a fondo el modelo con características más densas como adecuación o fluidez.

### 1.4.3 Métricas basadas en Deep learning

Se están empleando métodos más innovadores como el DL o las redes neuronales. Por ejemplo, se han empleado para escoger la traducción más acertada entre las hipotéticas comparando estas a una de referencia.

Un ejemplo sería la métrica **UoW-LSTM** que está basada en espacios de vector densos y LSTM (Long Short Term Memory), que son tipos de RNN (Redes Neuronales Recurrentes).

### 1.4.4 Métrica de evaluación BLEU

Se explicará la métrica de evaluación de **BLEU** más exhaustivamente. [24]

**BLEU** se propuso en el año 2002 y sigue siendo hoy en día una de las métricas de referencia a la hora de evaluar las traducciones automáticas. Buscaba solucionar problemas de las métricas humanas como el coste y el tiempo empleado.

Para evaluar la calidad de una traducción, se mide lo cerca que está de una o varias de referencia según una métrica numérica. Por tanto, se necesita:

- Una métrica de similitud numérica
- Un corpus de traducciones de referencia humanas de calidad

Se inspiró en la WER previamente mencionada. La idea principal era emplear una media ponderada de coincidencias de frases de distintas longitudes con las traducciones de referencia. Las distintas ponderaciones dan lugar a distintas versiones de la métrica. En este trabajo, se describe una de base.

Teniendo en cuenta que hay más de una traducción correcta, empleando distintos sinónimos y órdenes en la oración, los humanos distinguimos fácilmente una buena traducción de una mala.

En este ejemplo de dos traducciones automáticas de una oración en chino vemos claramente una diferencia en la calidad.

Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party.

Candidate 2: It is to insure the troops forever hearing the activity guidebook that party direct.

**ILUSTRACIÓN 14. Ejemplo de dos oraciones candidatas para valoración BLEU. [24]**

Comparamos con tres traducciones realizadas por traductores humanos.

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.

Reference 3: It is the practical guide for the army always to heed the directions of the party.

**ILUSTRACIÓN 15. Ejemplo de tres oraciones de referencia para valoración BLEU. [24]**

Observamos que la traducción más correcta (Candidate 1) comparte muchas palabras y grupos de ellas con las traducciones de referencia.

Para medir la **precisión**, se cuenta el número de n-gramas de la traducción candidata que coinciden con n-gramas de cualquiera de las traducciones de referencia y después se divide entre el número de palabras en la traducción candidata.

$$Precisión = \frac{Nº\ ngramas\ de\ candidata\ en\ cualquiera\ de\ las\ referencias}{nº\ total\ de\ palabras\ en\ candidata}$$

Los n-gramas son secuencias de palabras en los que **n** representa el número de palabras. Por ejemplo, en la oración “La vida sigue y hay que procurar olvidar.”:

| unigrama | bigrama | trigrama |
|---|---|---|
| La | La vida | La vida sigue |
| vida | vida sigue | vida sigue y |
| sigue | sigue y | sigue y hay |
| y | y hay | y hay que |
| hay | hay que | hay que procurar |
| que | que procurar | que procurar olvidar |
| procurar | procurar olvidar | |
| olvidar | | |

**TABLA 1. Representación de una oración en unigramas, bigramas y trigramas.**

El problema es que se dan casos en los que se generan demasiadas palabras "razonables" y el resultado es una traducción con precisión alta, pero de poca calidad como se muestra en el ejemplo siguiente.

Candidate: the the the the the the the.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

Modified Unigram Precision = 2/7.[3]

**ILUSTRACIÓN 16. Muestra de falso positivo en BLEU. [24]**

Para solucionar este problema se emplean los n-gramas modificados de precisión.

1. Se cuenta el número máximo de veces que ocurre una palabra en cualquiera de las traducciones de referencia (*Max_Ref_Count*).
2. Se toma la cuenta total de cada palabra de la traducción candidata (*Count*) y se limita al máximo de las de referencia (*Count clip*).
3. Se divide entre el total de palabras de la candidata.

$$Count_{clip} = min(Count, Max\_Ref\_Count)$$

Para el ejemplo 1 obtendríamos los siguientes resultados con unigramas:

| Cuenta candidato | Count | Ref1 Count | Ref2 Count | Ref3 Count | Max Ref Count | Clip Count |
|---|---|---|---|---|---|---|
| **It** | 1 | 1 | 1 | 1 | 1 | 1 |
| **is** | 1 | 1 | 1 | 1 | 1 | 1 |
| **a** | 1 | 1 | 0 | 0 | 1 | 1 |
| **guide** | 1 | 1 | 0 | 0 | 1 | 1 |
| **to** | 1 | 1 | 0 | 1 | 1 | 1 |
| **action** | 1 | 1 | 0 | 0 | 1 | 1 |
| **which** | 1 | 0 | 1 | 0 | 1 | 1 |
| **ensures** | 1 | 1 | 0 | 0 | 1 | 1 |
| **that** | 2 | 2 | 0 | 0 | 2 | 2 |
| **the** | 3 | 1 | 4 | 4 | 4 | 3 |
| **military** | 1 | 1 | 1 | 0 | 1 | 1 |
| **always** | 1 | 0 | 1 | 1 | 1 | 1 |
| **obeys** | 0 | 0 | 0 | 0 | 0 | 0 |
| **commands** | 1 | 1 | 0 | 0 | 1 | 1 |
| **of** | 0 | 0 | 1 | 1 | 1 | 0 |
| **party** | 1 | 0 | 0 | 1 | 1 | 1 |
| **18** | | | | | | **17** |

**TABLA 2. Resultados de los unigramas para una oración de ejemplo.**

Mientras que para el ejemplo 2 obtendríamos:

| Cuenta candidato | Count | Ref1 Count | Ref2 Count | Ref3 Count | Max Ref Count | Clip Count |
|---|---|---|---|---|---|---|
| **It** | 1 | 1 | 1 | 1 | 1 | 1 |
| **is** | 1 | 1 | 1 | 1 | 1 | 1 |
| **to** | 1 | 1 | 0 | 1 | 1 | 1 |
| **Insure** | 1 | 0 | 0 | 0 | 0 | 0 |
| **The** | 2 | 1 | 4 | 4 | 4 | 2 |
| **Troops** | 1 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Forever** | 1 | 1 | 0 | 0 | 1 | 1 |
| **Hearing** | 1 | 0 | 0 | 0 | 0 | 0 |
| **Activity** | 1 | 0 | 0 | 0 | 0 | 0 |
| **Guidebook** | 1 | 0 | 0 | 0 | 0 | 0 |
| **That** | 1 | 2 | 0 | 0 | 2 | 1 |
| **Party** | 1 | 1 | 1 | 1 | 1 | 1 |
| **Direct** | 1 | 0 | 0 | 0 | 0 | 0 |
| **14** | | | | | | **8** |

**TABLA 3. Cuenta de n-gramas en la oración de ejemplo.**

Dependiendo del valor de n en el n-grama escogido, se estudia la adecuación (unigramas) o la fluencia (n-gramas de mayor tamaño).

En el caso de múltiples oraciones, se emplea la puntuación modificada de precisión. Esta contabiliza los n-gramas de cada oración y a continuación divide los n-gramas modificados entre el total de n-gramas candidatos.

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

**Penalización de brevedad de la oración**

Las oraciones traducidas no deberían ser demasiado largas ni demasiado cortas. Los n-gramas de precisión penalizan las oraciones demasiado largas, pero no aquellas que son demasiado cortas. Por ello se emplea una penalización de brevedad en forma de factor multiplicativo. De esta forma, una oración candidata tendrá que ser equivalente a las de referencia en longitud, elección de palabras y orden de estas. Para no perjudicar gravemente las variaciones en longitud de las oraciones más breves, se tiene en cuenta el corpus al completo. Se toma la longitud total del corpus de pruebas, *r*, tomando las longitudes que mejor se ajustan a cada oración candidata. La penalización de brevedad se toma como una exponencial de *r/c* donde *c* es la longitud del corpus candidato.

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

Finalmente, aplicando la penalización de brevedad y empleando los n-gramas de precisión se llega a la métrica BLEU.

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

El comportamiento es más aparente en el dominio logarítmico,

$$\log BLEU = min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{N} w_n \log p_n$$

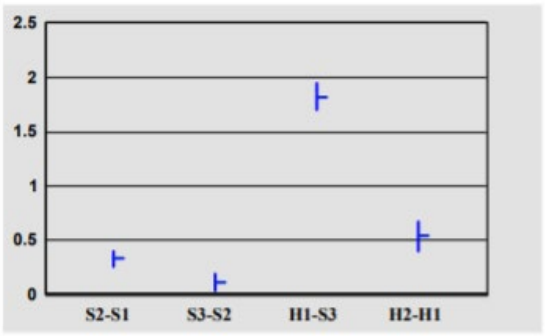
*N* se toma generalmente como 4 y los pesos como $w_n = {}^{1}/_{N}$.

**Evaluación BLEU**

La métrica BLEU tiene un rango que va de 0 a 1. Es necesario apuntar que una puntuación de 1 es muy difícil de obtener a menos que se trate de una copia idéntica a la oración de referencia. También cabe destacar que cuantas más oraciones de referencia haya, mayor será la puntuación obtenida.

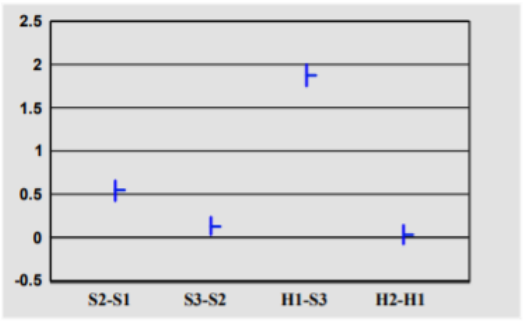
Se realizó una evaluación humana con 10 jueces hablantes nativos de inglés (se llamarán en adelante, juicios monolingües) y 10 hablantes nativos de chino que vivían en EE. UU. y hablaban inglés (se llamarán en adelante, juicios bilingües). Tenían que evaluar un total de 250 oraciones traducidas del chino al inglés con valores del 1 (muy malo) al 5 (muy bueno).

Los resultados mostraron los siguientes resultados:




| | | | | |
|---|---|---|---|---|
| +95% | 0.400 | 0.194 | 1.945 | 0.670 |
| -95% | 0.252 | 0.034 | 1.705 | 0.400 |
| monolingual | 0.326 | 0.114 | 1.825 | 0.535 |

**ILUSTRACIÓN 17. Juicios monolingües.**




| | | | | |
|---|---|---|---|---|
| +95% | 0.667 | 0.238 | 2.007 | 0.145 |
| -95% | 0.435 | 0.042 | 1.759 | -0.069 |
| bilingual | 0.551 | 0.140 | 1.883 | 0.038 |

**ILUSTRACIÓN 18. Juicios bilingües.**

**BLEU vs Evaluación humana**

La figura muestra la regresión lineal (LR) de las evaluaciones del grupo monolingüe en función de la evaluación BLEU de dos oraciones. El resultado es de un coeficiente de correlación de 0.99 que indica que BLEU capta el juicio humano de manera correcta.

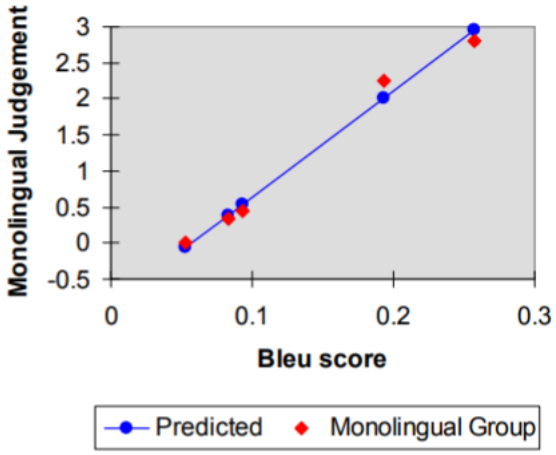



**ILUSTRACIÓN 19. Predicciones de BLEU de juicios monolingües.**

Para el grupo bilingüe se obtiene un coeficiente de 0.96.

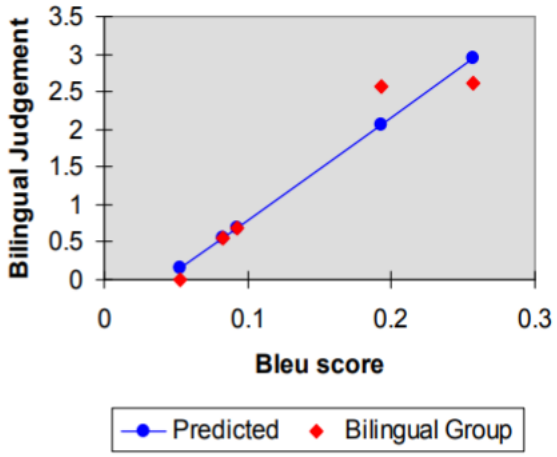



**ILUSTRACIÓN 20. Predicciones de BLEU de juicios bilingües.**

Se tomó el sistema peor valorado como punto de referencia y se compararon las evaluaciones BLEU con las evaluaciones humanas de los sistemas restantes. Se normalizaron linealmente los resultados y se obtuvo la siguiente gráfica:

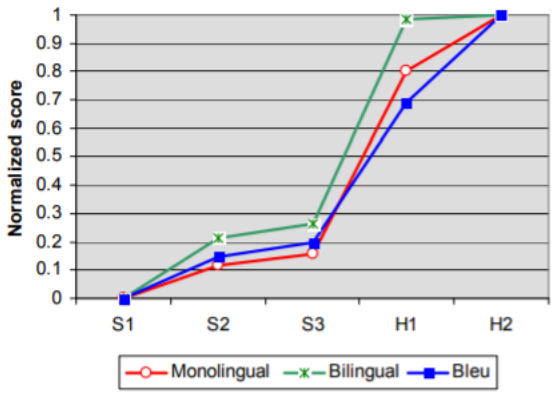



**ILUSTRACIÓN 21. Representación de juicios monolingües, bilingües y BLEU.**

La figura ilustra la alta correlación que existe entre la evaluación BLEU y la monolingüe. También muestra claramente la calidad superior de las traducciones humanas frente a las automáticas.

### 1.4.5 Métricas de evaluación automática para MT hoy

Las técnicas de traducción automáticas han evolucionado enormemente en los últimos años, obteniendo hoy traducciones de alta calidad que se emplean en todo tipo de entornos. Sin embargo, la tarea de la evaluación automática ha quedado relegada a un segundo plano. Hoy en día se continúan empleando métricas que precisan de una traducción de referencia generada manualmente para obtener la similitud entre esta y la traducción generada por la MT. Métricas como BLEU, METEOR o ROUGE, que se centran en similitudes en las características léxicas, siguen siendo las más empleadas a la hora de evaluar traducciones automáticas dada su rapidez y su facilidad de uso. Estas métricas a pesar de su facilidad de empleo no son efectivas a la hora de medir la calidad absoluta de una traducción, pues carecen de información sobre la semántica de las oraciones estudiadas.

Sin embargo, aunque no sea el foco de atención- solo hubo 24 sumisiones en la tarea de métricas en la WMT de 2019 [25] frente a las 153 de MT.- se están desarrollando métricas de evaluación avanzadas que tienen en cuenta aspectos que van más allá del léxico.

# Capítulo 2. Marco Teórico

## 2.1 Traducción automática: Microsoft Azure Translator

En este trabajo se ha empleado la herramienta de traducción de Microsoft, llamada Azure. En la página de Microsoft, se describe el servicio de la siguiente manera: "un servicio de traducción automática basado en la nube que forma parte de la familia de API de Azure Cognitive Services utilizada para crear aplicaciones inteligentes".

Esta herramienta emplea desde hace años tecnología NLP en lugar de SLP, ya que obtiene resultados de mayor calidad.

Hay diferentes opciones de traducción dentro de Azure:

- Traducción de texto
- Traducción de documentos
- Traducción personalizada

En este caso, se ha empleado la opción de traducción de texto mediante una API de REST.

### 2.1.1 API

Una API es una Interfaz de Programación de Aplicaciones (en inglés Application Programming Interface), que permite a dos aplicaciones interactuar entre ellas mediante un software que hace de intermediario. Las APIs son ampliamente utilizadas hoy en día, ya que permite que sus productos y servicios se comuniquen con otros, sin la implementación explícita de los mismos, mediante el empleo de solicitudes de procedimiento y devoluciones de datos. Simplifica enormemente el desarrollo de aplicaciones y potencia la innovación de la tecnología.

La herramienta de traducción empleada para este proyecto se basa en la conexión de una arquitectura de aplicaciones de microservicios a través de API, a la que se denomina aplicación basada en la nube.

**REST API**

Una API tipo REST (Representational State Transfer o Transferencia de Estado Representacional en español) es una API que emplea unos límites de arquitectura concretos. Una API de REST obtiene la información en formato HTTP y genera respuestas en diferentes formatos como JSON (JavaScript Object Notation) o XLT, siendo JSON el más popular. Algunas de las características de las API REST son:

- **Gestión de solicitudes a través de HTTP**. Las operaciones más importantes son POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- **Comunicación entre cliente y servidor sin estado**. La información del cliente no se almacena entre solicitudes, es decir, cada una es independiente y está desconectada del resto.
- **Sistema de capas**. Arquitectura jerárquica entre los componentes.

- **Uso de hipermedios**. Se proporcionan al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

La API REST del caso concreto de Azure Translator para traducción de texto, nos muestra los siguientes métodos disponibles al servicio del cliente:

| | | |
|---|---|---|
| languages | GET | Devuelve los idiomas soportados en el momento para las demás opciones. |
| translate | POST | Traduce texto del idioma fuente especificado al idioma objetivo. |
| transliterate | POST | Convierte una palabra o frase del idioma de origen al idioma objetivo en función de su similitud fonética. |
| detect | POST | Identifica el idioma de origen. |
| breaksentence | POST | Devuelve un array con la longitud de las oraciones en el texto de origen. |
| dictionary/lookup | POST | Devuelve alternativas para traducciones de palabras simples. |
| dictionary/examples | POST | Devuelve como un término es utilizado en contexto. |

**TABLA 4. Diferentes llamadas a la API de Azure.**

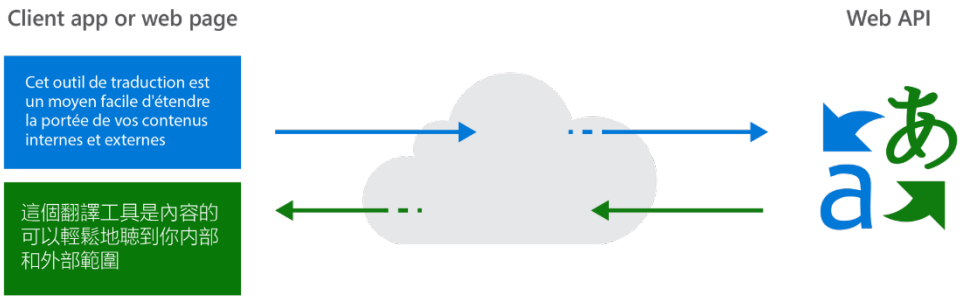



**ILUSTRACIÓN 22 Ilustración representativa del uso de la API de Azure Translator.**[18]

Para utilizar la herramienta de traducción, se crea una cuenta de suscripción a Azure y posteriormente se crea un recurso en Translator para obtener una clave y un punto de conexión.

[18] https://azure.microsoft.com/en-us/services/cognitive-services/translator/#overview

Se realiza la siguiente llamada a la API, donde se indica la clave, el idioma de origen y el idioma destino, la región desde donde se realiza la llamada y se indica que se active la detección de vulgaridades.

```python
# -*- coding: utf-8 -*-
import os, requests, uuid, json

key_var_name = 'TRANSLATOR_TEXT_SUBSCRIPTION_KEY'
if not key_var_name in os.environ:
    raise Exception('Please set/export the environment variable: {}'.format(key_var_name))
subscription_key = os.environ[key_var_name]

endpoint_var_name = 'TRANSLATOR_TEXT_ENDPOINT'
if not endpoint_var_name in os.environ:
    raise Exception('Please set/export the environment variable: {}'.format(endpoint_var_name))
endpoint = os.environ[endpoint_var_name]

path = '/translate?api-version=3.0'
# params = '&from=en&to=es'
# params = '&from=en&to=zh-Hans'
params = '&from=en&to=es&profanityAction=Marked&profanityMarker=Tag'
constructed_url = endpoint + path + params

headers = {
    'Ocp-Apim-Subscription-Key': subscription_key,
    'Ocp-Apim-Subscription-Region': 'westeurope',
    'Content-type': 'application/json',
    'X-ClientTraceId': str(uuid.uuid4())
}
```

**ILUSTRACIÓN 23. Script de llamada a la API para traducción de la base de datos.**

Una vez completada la traducción, se obtiene un documento en formato CSV con todas las traducciones. Un archivo CSV (comma separated values), es cualquier archivo de texto para representar datos en forma de tabla en el que los caracteres están separados por comas.

## 2.2 Conjuntos de datos

A la hora de trabajar en el campo de la AI, es esencial contar con bases de datos de calidad. En este caso el objetivo es el entrenamiento de un asistente de diálogo, por lo que es necesario emplear un conjunto de datos que represente de manera fehaciente las interacciones que suceden entre seres humanos.

Las bases de datos que se podían encontrar hasta mediados de la década del 2010 estaban generalmente centradas en un tema concreto y resultaban adecuadas para tareas específicas, pero no tenían interés para usos más generales.

A partir de 2016, los agentes conversacionales de dominio abierto (open-domain) comenzaron a desarrollarse y con ello, bases de datos más completas para suplir las carencias de los conjuntos precedentes. Este nuevo enfoque de los sistemas conversacionales trata de abordar tres desafíos: semántica (no solo entiende el contenido del diálogo, sino que identifica las necesidades emocionales durante la conversación), consistencia (el sistema debe demostrar una “personalidad” consistente para ganar la confianza del usuario) e interactividad (la capacidad de generar respuestas interpersonales) [26]. Algunas de las más relevantes son Daily Dialogue [27], Empathetic

Dialogues [28], PersonaChat [29] o Wizard of Wikipedia [30]. Incluso se están desarrollando sistemas conversacionales que, partiendo de una base, continúan aprendiendo mediante la propia interacción conversacional como LIGHT WILD [31].

Para este trabajo, se van a emplear los conjuntos de datos Daily Dialog y Empathetic Dialogues.

## 2.2.1 Daily Dialog

La base de datos de Daily Dialog, consiste en un conjunto de interacciones multi-turno. Estas oraciones se han recolectado de páginas web educativas para el aprendizaje del inglés, están escritas por humanos y constituyen un reflejo de la comunicación diaria que lleva a cabo el ser humano.

En contraposición a las bases de datos preexistentes, unas, de dominio específico, y otras, demasiado dispersas como para capturar el tema central de la conversación, se desarrolla Daily Dialog, en adelante DD.

### 2.2.1.1 Construcción de la base de datos

Se emplean datos de diferentes páginas web que sirven para ayudar a los estudiantes de inglés a practicar diálogos del día a día. Comparando con otras bases de datos:

- Los datos de DD están escritos por personas con la intención del aprendizaje, lo que los hacen más formales que otras bases de datos basadas en interacciones en redes sociales.
- Generalmente las conversaciones de DD se centran en un tema o contexto físico.
- Normalmente tienen un número razonable de turnos de palabra (aproximadamente ocho) en una conversación a diferencia de bases de datos como OpenSubtitles, que pueden tener más de ciento cincuenta turnos de palabra en una conversación.

### 2.2.1.3 Características

En la siguiente tabla se muestran las características básicas sobre DD:

| | |
|---|---|
| Número total de diálogos | 13.118 |
| Media de turnos de palabra por diálogo | 7,9 |
| Media de palabras por diálogo | 114,7 |
| Media de palabras por intervención | 14,6 |

**TABLA 5. Características básicas de DD.**

Además de las características básicas, se describirán a continuación otras características de la base de datos.

- **Temas diarios**

  Se dividen en 10 categorías siendo aquellas más abundantes: relaciones (33,33%), vida ordinaria (28,36%) y trabajo (14,49%).

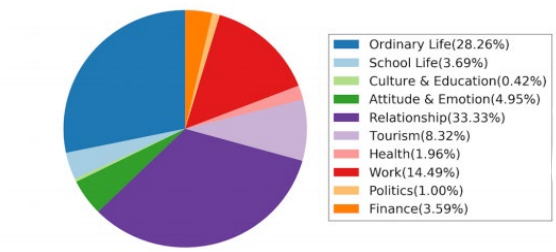



**ILUSTRACIÓN 24 Distribución temas. [27]**

- **Flujo de diálogo bidireccional**

  Siguen un flujo de diálogo natural. Comparado con otras bases de datos como comentarios de Twitter o Reddit, que se basan en subir contenido y responder al mismo, y pueden resultar en flujos de diálogo ambiguos, DD es más consistente con la manera diaria de comunicación.

  Se refleja en flujos de diálogo de pregunta-información o en los que son directivos-comisivos. En la siguiente tabla se muestra la distribución de los cuatro actos del habla:

| Informativos | Interrogativo | Directivos | Comisivos |
|---|---|---|---|
| 46,532 | 29,428 | 17,295 | 9,724 |
| 45,2% | 28,6% | 16,8% | 9,4% |

**TABLA 6. Estadística de intención de DD**

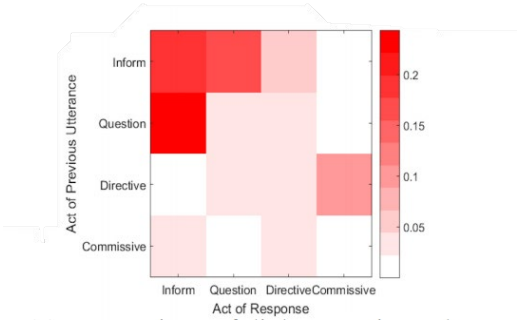



**ILUSTRACIÓN 25. Interacciones de los actos del diálogo en parejas de intervenciones. [27]**

- **Patrones de comunicación certeros**

  Además de los patrones básicos *preguntas-informaciones* y *directivas-comisivas* que son flujos de diálogo bi-turno, en DD se encuentran dos patrones de flujo multi-turno únicos:

  - **Patrón 1:** En la comunicación humano a humano, las personas se sienten inclinadas a contestar las preguntas y después iniciar una nueva pregunta para que el diálogo dure. El hablante pasa de proveedor de información a buscador de información en el mismo turno de palabra. El 18,3% de los diálogos en DD muestran este patrón.
  - **Patrón 2:** Cuando alguien hace una propuesta o sugerencia, el otro hablante generalmente aporta otra idea. Esto resulta en patrones del tipo *directivo-directivo-comisivo*. Representan un 9,2% de los patrones de diálogo de DD.

- **Rica en emociones**

  Otro de los objetivos principales de la base de datos es representar la carga emocional que se muestra en la comunicación humana. Debido a la dificultad de la clasificación emocional automática, se han etiquetado manualmente las intervenciones. Se muestra debajo una tabla con las diferentes etiquetas emocionales y sus proporciones estadísticas:

| | *Cuenta* | *De EU*[19] | *Del total* |
|---|---|---|---|
| *Enfado* | 1022 | 5,87 | 0,99 |
| *Asco* | 353 | 2,03 | 0,34 |
| *Miedo* | 74 | 1,00 | 0,17 |
| *Felicidad* | 12885 | 74,02 | 12,51 |
| *Tristeza* | 1150 | 6,61 | 1,12 |
| *Sorpresa* | 1823 | 10,47 | 1,77 |
| *Otras* | 85572 | - | 83,10 |

**TABLA 7. Estadísticas de emociones en DD.**

[19] EU se refiere a intervenciones que presentan una emoción dentro de las seis principales de Eckman [32].

### 2.2.1.3 Disposición

La base de datos obtenida tras la traducción de DD, contiene varios documentos CSV, de los que se van a emplear:

- DAILYD_translation_en2es.csv. Contiene cuatro columnas:
    - UID → Contiene la codificación de las intervenciones con la siguiente forma:

      **DAILYD-000000-0000**

      Donde el primer elemento es DAILYD, el nombre de la base de datos, el segundo es el código del diálogo y el tercero el de la intervención dentro del diálogo.

    - SEG → Contiene las intervenciones originales en inglés.
    - translation → Contiene las intervenciones traducidas al español.
    - profanity → Traducido como vulgaridad al español, indica si existe alguna en la intervención.

| | UID | SEG | translation | profanity |
|---|---|---|---|---|
| **0** | DAILYD-000000-0000 | The kitchen stinks . | La cocina apesta. | None |
| **1** | DAILYD-000000-0001 | I'll throw out the garbage . | Voy a tirar la basura . | None |
| **2** | DAILYD-000001-0000 | So Dick , how about getting some coffee for to... | Dick , ¿qué tal tomar un café para esta noche? | None |
| **3** | DAILYD-000001-0001 | Coffee ? I don ' t honestly like that kind of... | ¿Café? Honestamente no me gusta ese tipo de c... | None |
| **4** | DAILYD-000001-0002 | Come on , you can at least try a little , bes... | Vamos, al menos puedes probar un poco, además... | None |
| **...** | ... | ... | ... | ... |
| **102963** | DAILYD-013116-0010 | Well , thank you very much for all that infor... | Bueno, muchas gracias por toda esa informació... | None |
| **102964** | DAILYD-013116-0011 | Are you going to make an offer today ? | ¿Vas a hacer una oferta hoy? | None |
| **102965** | DAILYD-013116-0012 | Yes . My customer is in urgent need of the st... | Sí. Mi cliente está en necesidad urgente de l... | None |
| **102966** | DAILYD-013116-0013 | Ok , I'll get this rate right away . | Ok , Voy a obtener esta tarifa de inmediato. | None |
| **102967** | DAILYD-013116-0014 | Thank you . | Gracias. | None |

102968 rows × 4 columns

**ILUSTRACIÓN 26. Representación del archivo DAILYD_translation_en2es.**

- DAILYD_translation_es2en.csv. Contiene cuatro columnas:
    - UID → Contiene la codificación de las intervenciones con la siguiente forma:

      **DAILYD-000000-0000**

Donde el primer elemento es DAILYD, el nombre de la base de datos, el segundo es el código del diálogo y el tercero el de la intervención dentro del diálogo.

- SEG → Contiene las intervenciones traducidas al español.
- translation → Contiene las intervenciones traducidas de vuelta al inglés del español.
- profanity → Traducido como vulgaridad al español, indica si existe alguna en la intervención.

| | UID | SEG | translation | profanity |
|---|---|---|---|---|
| **0** | DAILYD-000000-0000 | La cocina apesta. | The kitchen sucks. | None |
| **1** | DAILYD-000000-0001 | Voy a tirar la basura . | I'm going to throw away the trash. | None |
| **2** | DAILYD-000001-0000 | Dick , ¿qué tal tomar un café para esta noche? | dick, how about having a coffee for tonight? | dick |
| **3** | DAILYD-000001-0001 | ¿Café? Honestamente no me gusta ese tipo de c... | Coffee? I honestly don't like that kind of th... | None |
| **4** | DAILYD-000001-0002 | Vamos, al menos puedes probar un poco, además... | Come on, at least you can try some, besides y... | None |
| **...** | ... | ... | ... | ... |
| **102963** | DAILYD-013116-0010 | Bueno, muchas gracias por toda esa informació... | Well, thank you so much for all that informat... | None |
| **102964** | DAILYD-013116-0011 | ¿Vas a hacer una oferta hoy? | Are you going to make an offer today? | None |
| **102965** | DAILYD-013116-0012 | Sí. Mi cliente está en necesidad urgente de l... | Yes. My client is in urgent need of steel pla... | None |
| **102966** | DAILYD-013116-0013 | Ok , Voy a obtener esta tarifa de inmediato. | Okay, I'm going to get this rate right away. | None |
| **102967** | DAILYD-013116-0014 | Gracias. | Thank you. | None |

102968 rows × 4 columns

**ILUSTRACIÓN 27. Representación del archivo DAILYD_translation_es2en.**

## 2.2.2 Empathetic Dialogues

La base de datos de Empathetic Dialogues, es un conjunto de conversaciones basadas en situaciones emocionales. Permite entrenar sistemas de diálogo que presentan mayor empatía según evaluadores humanos frente a sistemas entrenados con otras bases de datos, entrenadas con bases de datos de conversación de internet.

### 2.2.2.1 Construcción de la base de datos

Sa base de datos se recolectó mediante la plataforma ParlAI [33], una plataforma de software de código abierto para la investigación de diálogo implementada en Python. Esta plataforma tiene integrada Amazon Mechanical Turk (MTurk[20]), una plataforma de crowdsourcing que oferta trabajos simples de bajo coste que requieren de una inteligencia humana. Mediante la mencionada plataforma, se contrató a 810

[20] https://www.mturk.com/

trabajadores nativos de Estados Unidos. Se emparejó a estos trabajadores y se les pidió que describieran una emoción y una situación en la que hubieran experimentado dicha emoción, y que tuvieran una conversación sobre la situación descrita. Para asegurar una cierta calidad, se realizaron comprobaciones en grupos aleatorios de conversaciones.

#### 2.2.2.2 Características

Las características básicas de la base de datos se representan en la siguiente tabla:

| | |
|---|---|
| Número total de diálogos | 24.850 |
| Media de turnos de palabra por diálogo (intervenciones) | 4,31 |
| Media de palabras por intervención | 15,2 |

**TABLA 8. Características básicas de ED.**

Además de estas características, la característica principal de ED es que tiene una fuerte carga emocional. Se consideran 32 etiquetas emocionales que se asignan a cada una de las conversaciones que conforman la base de datos. Estas etiquetas emocionales comprenden tanto emociones positivas como negativas.

En este proyecto, se han mapeado las 32 emociones a las 6 principales de Eckman [32].

| Emotion | Most-used speaker words | Most-used listener words | Training set emotion distrib |
|---|---|---|---|
| Surprised | got,shocked,really | that's,good,nice | 5.1% |
| Excited | going,wait,i'm | that's,fun,like | 3.8% |
| Angry | mad,someone,got | oh,would,that's | 3.6% |
| Proud | got,happy,really | that's,great,good | 3.5% |
| Sad | really,away,get | sorry,oh,hear | 3.4% |
| Annoyed | get,work,really | that's,oh,get | 3.4% |
| Grateful | really,thankful,i'm | that's,good,nice | 3.3% |
| Lonely | alone,friends,i'm | i'm,sorry,that's | 3.3% |
| Afraid | scared,i'm,night | oh,scary,that's | 3.2% |
| Terrified | scared,night,i'm | oh,that's,would | 3.2% |
| Guilty | bad,feel,felt | oh,that's,feel | 3.2% |
| Impressed | really,good,got | that's,good,like | 3.2% |
| Disgusted | gross,really,saw | oh,that's,would | 3.2% |
| Hopeful | i'm,get,really | hope,good,that's | 3.2% |
| Confident | going,i'm,really | good,that's,great | 3.2% |
| Furious | mad,car,someone | oh,that's,get | 3.1% |
| Anxious | i'm,nervous,going | oh,good,hope | 3.1% |
| Anticipating | wait,i'm,going | sounds,good,hope | 3.1% |
| Joyful | happy,got,i'm | that's,good,great | 3.1% |
| Nostalgic | old,back,really | good,like,time | 3.1% |
| Disappointed | get,really,work | oh,that's,sorry | 3.1% |
| Prepared | ready,i'm,going | good,that's,like | 3% |
| Jealous | friend,got,get | get,that's,oh | 3% |
| Content | i'm,life,happy | good,that's,great | 2.9% |
| Devastated | got,really,sad | sorry,oh,hear | 2.9% |
| Embarrassed | day,work,got | oh,that's,i'm | 2.9% |
| Caring | care,really,taking | that's,good,nice | 2.7% |
| Sentimental | old,really,time | that's,oh,like | 2.7% |
| Trusting | friend,trust,know | good,that's,like | 2.6% |
| Ashamed | feel,bad,felt | oh,that's,i'm | 2.5% |
| Apprehensive | i'm,nervous,really | oh,good,well | 2.4% |
| Faithful | i'm,would,years | good,that's,like | 1.9% |

**ILUSTRACIÓN 28. Distribución de las etiquetas de los datos de entrenamiento de ED con las 3 palabras más usadas por el interlocutor y el receptor. [28]**

### 2.2.1.3 Disposición

La base de datos obtenida tras la traducción contiene los siguientes documentos CVS:

- MPATHY_tranlation_en2es.csv. Contiene cuatro columnas:
    - UID → Contiene la codificación de las intervenciones con la siguiente forma:

      **MPATHY-000000-0000**

      Donde el primer elemento es MPATHY, el nombre de la base de datos, el segundo es el código del diálogo y el tercero el de la intervención dentro del diálogo.
    - SEG → Contiene las intervenciones originales en inglés.
    - translation → Contiene las intervenciones traducidas al español.
    - profanity → Traducido como vulgaridad al español, indica si existe alguna en la intervención.

| | UID | SEG | translation | profanity |
|---|---|---|---|---|
| **0** | MPATHY-000001-0000 | I remember going to see the fireworks with my ... | Recuerdo que iba a ver los fuegos artificiales... | None |
| **1** | MPATHY-000001-0001 | Was this a friend you were in love with, or ju... | ¿Era un amigo del que estabas enamorada o sólo... | None |
| **2** | MPATHY-000001-0002 | This was a best friend. I miss her. | Era un mejor amigo. La extraño. | None |
| **3** | MPATHY-000001-0003 | Where has she gone? | ¿Adónde se ha ido? | None |
| **4** | MPATHY-000001-0004 | We no longer talk. | Ya no hablamos. | None |
| **...** | ... | ... | ... | ... |
| **107215** | MPATHY-024832-0003 | I live in Texas to so i know those feels | Vivo en Texas para que yo sepa que esos sentim... | None |
| **107216** | MPATHY-024847-0000 | I have a big test on Monday, I am so nervous. | Tengo una gran prueba el lunes, estoy tan nerv... | None |
| **107217** | MPATHY-024847-0001 | What is the test on? | ¿En qué consiste la prueba? | None |
| **107218** | MPATHY-024847-0002 | It's for my Chemistry class. I haven't slept m... | Es para mi clase de Química. No he dormido muc... | None |
| **107219** | MPATHY-024847-0003 | Chemistry is quite difficult,have you studied ... | La química es bastante difícil, ¿has estudiado... | None |

**ILUSTRACIÓN 29. Representación del archivo MPATHY_translation_en2es.**

- MPATHY_tranlation_es2en.csv. Contiene cuatro columnas:
    - UID → Contiene la codificación de las intervenciones con la siguiente forma:

      **MPATHY-000000-0000**

      Donde el primer elemento es MPATHY, el nombre de la base de datos, el segundo es el código del diálogo y el tercero el de la intervención dentro del diálogo.

- SEG → Contiene las intervenciones traducidas al español.
- translation → Contiene las intervenciones traducidas al inglés de las intervenciones traducidas inicialmente al español.
- profanity → Traducido como vulgaridad al español, indica si existe alguna en la intervención.

| | UID | SEG | translation | profanity |
|---|---|---|---|---|
| **0** | MPATHY-000001-0000 | Recuerdo que iba a ver los fuegos artificiales... | I remember seeing the fireworks with my best f... | None |
| **1** | MPATHY-000001-0001 | ¿Era un amigo del que estabas enamorada o sólo... | Was he a friend you were in love with or just ... | None |
| **2** | MPATHY-000001-0002 | Era un mejor amigo. La extraño. | He was a best friend. I miss her. | None |
| **3** | MPATHY-000001-0003 | ¿Adónde se ha ido? | Where'd he go? | None |
| **4** | MPATHY-000001-0004 | Ya no hablamos. | We don't talk anymore. | None |
| **...** | ... | ... | ... | ... |
| **107215** | MPATHY-024832-0003 | Vivo en Texas para que yo sepa que esos sentim... | I live in Texas so I know those feelings | None |
| **107216** | MPATHY-024847-0000 | Tengo una gran prueba el lunes, estoy tan nerv... | I have a great test on Monday, I'm so nervous. | None |
| **107217** | MPATHY-024847-0001 | ¿En qué consiste la prueba? | What is the test? | None |
| **107218** | MPATHY-024847-0002 | Es para mi clase de Química. No he dormido muc... | It's for my chemistry class. I haven't slept m... | None |
| **107219** | MPATHY-024847-0003 | La química es bastante difícil, ¿has estudiado... | Chemistry is pretty hard, have you studied hard? | None |

**ILUSTRACIÓN 30. Representación del archivo MPATHY_translation_es2en.**

## 2.3 Mecanismos de evaluación automática

En este caso, se hará uso principalmente de dos mecanismos de evaluación automática: BLEU, como modelo de base, y de los modelos de Sentence Transformers, en adelante ST.

### 2.3.1 Sentence Transformers

Se trata de un marco de trabajo de Python centrado en el embedding de oraciones, texto e imagen de última generación. Está en constante crecimiento, pero surgió a raíz del estudio del embedding de oraciones empleando redes siamesas de BERT [34]. El marco de trabajo está basado en PyTorch y en Transformers y ofrece una enorme cantidad de modelos pre entrenados para diferentes tareas. Para este trabajo, se ha empleado una herramienta muy útil que es la similitud coseno entre los embeddings de los diferentes modelos.

#### 2.3.1.1 BERT

BERT [35] (Bidirectional Encoder Representations from Transformers) es un método de representación del lenguaje mediante preentrenamiento, que obtiene resultados de alto nivel para una gran cantidad de tareas de NLP. Fue desarrollado por los investigadores de Google AI Language en el año 2018, y marcó un antes y un después en el mundo del NLP.

BERT está pre entrenado como un modelo de "conocimiento amplio" con corpus enormes como Wikipedia. El modelo después se ajusta y emplea para diferentes tareas más específicas de NLP. BERT fue el primer sistema no supervisado y profundamente bidireccional (deeply bidirectional) para preentrenamiento de NLP.

No supervisado significa que BERT ha sido entrenado simplemente con corpus de texto.

Sin embargo, la gran innovación que supuso BERT se debe al hecho de que sea profundamente bidireccional. BERT se creó a raíz de trabajos que estaban surgiendo en el momento como ELMo[21] o ULMFit, pero estos modelos eran o unidireccionales, o superficialmente bidireccionales. Básicamente, los modelos unidireccionales, contextualizan cada palabra empleando las palabras a su izquierda (o derecha).

En la oración "I made a bank deposit", la representación unilateral de bank se basaría en "I made a" y no en "deposit". BERT, emplea tanto el contexto de la izquierda como el de la derecha: "I made a … deposit".

Para conseguirlo, se enmascara un 15% de las palabras de entrada y se hace uso del codificador Transformer (que es profundamente bidireccional) para después predecir las palabras enmascaradas.

*Entrada: the man went to the [MASK1] . he bought a [MASK2] of milk.*

*Etiquetas: [MASK1] = store; [MASK2] = gallon*

Además, está entrenado en las relaciones entre oraciones. Para ello, el modelo recibe parejas de oraciones, y aprende a predecir si la segunda sucede a la primera en el documento original.

| | |
|---|---|
| *Sentence A: the man went to the store.* | *Sentence A: the man went to the store.* |
| *Sentence B: he bought a gallon of milk.* | *Sentence B: penguins are flightless.* |
| *Label: IsNextSentence* | *Label: NotNextSentence* |

#### 2.3.1.2 PyTorch

Es un paquete de computación científico basado en Python gratuito y de código abierto. Es una de las plataformas más valoradas a la hora de investigar DL, ya que provee velocidad y flexibilidad. Se desarrolló principalmente por Facebook en el año

[21] https://allennlp.org/elmo

2016, y hoy en día softwares de DL como Tesla Autopilot, Pyro de Uber o Transformers de HugginFace, se han construido en PyTorch.

**ILUSTRACIÓN 31. Logotipo de PyTorch.**

### 2.3.1.3 Transformers

Transformers [22], es una librería desarrollada por HuggingFace que provee arquitecturas de propósito general (como BERT, RoBERTa o XLM) para comprensión del lenguaje natural (NLU) y para la generación del lenguaje natural (NLG) con más de 30 modelos pre entrenados en más de 100 idiomas. Soporta los espacios de trabajo de PyTorch, TensorFlow y Jax.

### 2.3.1.4 Similitud coseno

Es una métrica que se emplea para medir cómo de similares son los vectores dentro de un espacio vectorial. Se mide por el coseno del ángulo que forman los vectores y determina si estos apuntan en la misma dirección. Su uso en el ámbito del NLP se centra en medir la similitud entre documentos, oraciones, palabras o texto de cualquier tipo.

Cuanto menor es el ángulo entre dos vectores, mayor es la similitud. Como se puede ver en la ilustración, el primer ejemplo representa dos vectores similares; el segundo, dos vectores sin relación; el tercero, dos vectores opuestos.

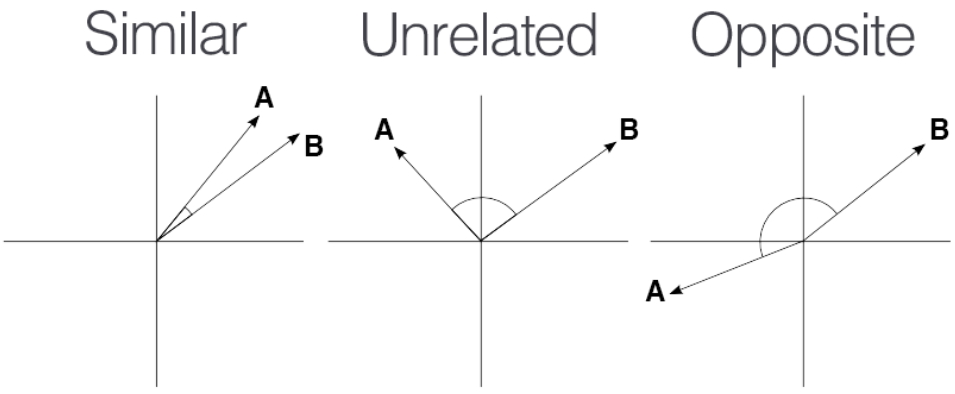



**ILUSTRACIÓN 32. Representaciones de dos vectores en un espacio vectorial. [23]**

[22] https://huggingface.co/transformers/
[23] https://medium.com/geekculture/cosine-similarity-and-cosine-distance-48eed889a5c4

Para calcular la similitud, se emplea la siguiente fórmula donde θ representa el ángulo entre los vectores *a* y *b* en un espacio multidimensional.

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

El resultado del cos(θ) se encuentra en el rango [-1,1]:

- -1 indicaría oposición de los vectores.
- 0 indicaría independencia.
- 1 indicaría similitud entre los vectores.

Ocasionalmente, se calcula la distancia coseno, que es la siguiente:

*Distancia coseno = 1- Similitud coseno*

Representa la "distancia" entre dos vectores. Si dos vectores tienen una similitud coseno de 1 (son perfectamente iguales), la distancia entre estos es igual a 0.

### 2.3.1.5 Coeficiente de Pearson

El coeficiente de Pearson[24] es una medida de dependencia lineal entre dos variables aleatorias cuantitativas. Se define con la siguiente fórmula:

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$$

Donde $\sigma_{XY}$ es la covarianza de (X,Y), $\sigma_X$ es la desviación estándar de la variable X y $\sigma_Y$ es la desviación estándar de la variable Y.

### 2.3.1.6 Modelos monolingües y modelos multilingües

La mayor parte de los modelos de ST son monolingües (generalmente en inglés). Estos modelos no se pueden emplear directamente para realizar la evaluación de la similitud entre oraciones en diferentes idiomas. Tampoco se pueden comparar los embedding realizados por modelos monolingües en diferentes idiomas, pues se mapean a espacios vectoriales diferentes. Para obtener modelos multilingües, se deben ajustar los modelos monolingües con datos lingüísticos cruzados (cross-lingual data) asegurando que ambos mapean en el mismo espacio vectorial.

ST presentaron en 2020 modelos multilingües [36], obtenidos mediante knowledge distillation (destilación del conocimiento). Para ello, se obtuvieron embeddings de oraciones del idioma de origen y se entrenaron los nuevos sistemas con oraciones

[24] https://es.wikipedia.org/wiki/Coeficiente_de_correlaci%C3%B3n_de_Pearson

traducidas y corpus paralelos, para imitar esos embeddings. Comparado con otros métodos, resulta sencillo extender el modelo con un número reducido de ejemplos a una lengua nueva y es más sencillo asegurar que ambos modelos se encuentran en el mismo espacio vectorial.

## 2.4 Librerías empleadas

### 2.4.1 NLTK

La librería Natural Language Toolkit (Conjunto de herramientas de lenguaje natural en español), es ampliamente utilizada en el mundo del NLP para construir programas en Python. Proporciona interfaces sencillas de utilizar para corpus y recursos léxicos como WordNet, una de las más empleadas. Además, contiene una gran cantidad de librerías para clasificación, tokenización, etiquetado y otras utilidades a la hora de manejar datos de lenguaje.

Dentro de la librería, en este trabajo se ha empleado la clase *bleu_score*. Este paquete permite obtener valoraciones de BLEU de manera sencilla.


**ILUSTRACIÓN 33. Logotipo de NLTK.**

### 2.4.2 Pandas

La librería Pandas, es una de las librerías más empleadas en Python y sirve para el análisis y manipulación de datos. Ofrece estructuras de datos y operaciones para manipular tablas numéricas. Pandas permite la importación de datos desde formatos de diferentes tipos como son JSON, CSV, o Excel.


**ILUSTRACIÓN 34. Logotipo de Pandas.**

### 2.4.3 NumPy

Numpy (Numerical Python en inglés) es una librería de tipo matemático, contiene todo tipo de operaciones matemáticas y lógicas para operar con vectores. Numpy permite crear vectores y matrices de gran tamaño y multidimensionales y es un software de código abierto que está en constante crecimiento.


**ILUSTRACIÓN 35. Logotipo de NumPy.**

### 2.4.4 SciPy

Scipy es una librería de código abierto para Python. Contiene herramientas y algoritmos matemáticos. Sirve para resolver todo tipo de problemas matemáticos, científicos, técnicos y relacionados con la ingeniería.


**ILUSTRACIÓN 36. Logotipo de SciPy.**

### 2.4.5 Matplotlib

Matplotlib[25] es una librería para crear elementos visuales estáticos, animados e interactivos en Python. Permite crear gráficos de manera sencilla y totalmente personalizable.


**ILUSTRACIÓN 37. Logotipo de Matplotlib.**

### 2.4.6 Seaborn

Searborn[26] es una librería de visualización de datos para Python basada en Matplotlib. Proporciona una interfaz de alto nivel para diseñar gráficos estadísticos informativos y atractivos.


**ILUSTRACIÓN 38. Logotipo de Seaborn.**

[25] https://matplotlib.org/
[26] https://seaborn.pydata.org/

### 2.4.7 Scikit-learn

Scikit-learn [27] o sklear es una librería de Python centrada en el ML. Está construida sobre NumPy, SciPy y matplotlib. Contiene herramientas eficientes y simples para el análisis predictivo de datos.


**ILUSTRACIÓN 39. Logotipo de Scikit-learn**

## 2.5 Entorno de programación: Google Colaboratory

Toda la programación de este proyecto se ha llevado a cabo en la plataforma de Colaboratory [28] o "Colab" de Google. Se trata de un servicio alojado en Jupyter Notebook que permite a cualquier usuario de Google escribir y ejecutar código de Python en el navegador mediante una máquina virtual dedicada a cada cuenta. Es apropiado para tareas de AI y análisis de datos. Los cuadernos creados con Colab se ejecutan en los servidores en la nube de Google y se almacenan en Google Drive o en Github, siendo sencillo compartirlos con otros usuarios.


**ILUSTRACIÓN 40. Logotipo de Google Colab**

### 2.5.1 Jupyter

Jupyter [29] Notebook es una aplicación web de código abierto que permite crear y compartir documentos con código y ecuaciones interactivas, intercaladas con texto. Los usos más comunes son manejo de datos, simulación numérica, modelado estadístico o aprendizaje automático. Se trata de una herramienta popular en la elaboración de artículos en el campo de la investigación científica.

[27] https://scikit-learn.org/stable/
[28] https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index#
[29] https://jupyter.org/


**ILUSTRACIÓN 41. Logotipo de Jupyter.**

### 2.5.2 GitHub

GitHub es una plataforma de desarrollo colaborativo que aloja código tanto abierto, accesible a cualquier desarrollador, como proyectos privados. Se basa en el sistema de control de versiones Git, con el que los desarrolladores pueden administrar su proyecto y tener un control de todas las versiones, guardándose una copia de cada versión nueva que se crea de un proyecto. Esto es especialmente útil para la comparación de versiones y el desarrollo de distintas ramas dentro del mismo proyecto. Uno de los puntos fuertes de la plataforma es su cualidad colaborativa, ya que permite a cualquier usuario comentar, mejorar y colaborar con aquellos proyectos que sean de código abierto.


**ILUSTRACIÓN 42. Logotipo de Github.**

# Capítulo 3. Desarrollo del proyecto

En este apartado se comprobará la calidad de las traducciones obtenidas con Azure Translator con varios métodos. Como se ha comentado previamente, por un lado, se ha empleado la métrica BLEU, que es la base de referencia a la hora de comprobar la calidad de traducciones y, por otro lado, se ha utilizado la tecnología de ST, una de las más avanzadas que se encuentran disponibles hoy en día.

Al ser las intervenciones traducidas la base de estudio de este proyecto, es necesario aclarar cómo se nombrará cada tipo de intervención a lo largo del capítulo:

- Intervención original. Son las intervenciones originales en inglés de las bases de datos.
- Intervención traducida. Se trata de las intervenciones originales traducidas al español.
- Intervención candidata. Se trata de las intervenciones traducidas que se han traducido de vuelta al inglés.

## 3.1 Obtención valoraciones BLEU

### 3.1.1 Preprocesamiento de los datos

Los datos empleados para el desarrollo del proyecto fueron previamente traducidos mediante la herramienta de traducción de Microsoft Azure por el Centro de Automática y Robótica de la ETSII.

Los resultados obtenidos están preparados para su uso sin necesidad de un procesado posterior ya que se emplea habitualmente como traductor de sitios web y similares.

En este caso, para poder evaluar correctamente las traducciones es necesario realizar un preprocesamiento de los datos para emplear las métricas de evaluación pertinentes.

Al realizar las primeras pruebas con la métrica de evaluación BLEU, se obtuvieron resultados muy bajos, que no correspondían a los valores esperados. Al realizar una revisión de los datos, se observó que los datos originales en inglés no compartían la misma forma que los datos de la segunda traducción.

Como se puede leer en [35], la importancia de los signos de puntuación y el rigor de estos depende de la lengua que se está estudiando. Los puntos o las comas se pueden emplear para separar palabras mientras que apóstrofos o guiones se pueden emplear para juntarlas. Esto es especialmente relevante en el caso de este trabajo, ya que el inglés emplea constantemente los signos de puntuación para unir palabras como se muestra en el siguiente ejemplo.


**ILUSTRACIÓN 43. Ejemplo del uso del apóstrofo.**

Para Leusch [37], al evaluar automáticamente traducciones al inglés, se debe decidir qué se considera como palabra independiente. Estudia para ello cuatro métodos de tokenización (separación del texto en oraciones y palabras).

- **Candidata original**
  Powell said: "We'd not be alone; that's for sure."
- **Eliminación de los signos de puntuación**
  Powell said We d not be alone that s for sure
- **Tokenización de los signos de puntuación**
  Powell said : " We'd not be alone ; that's for sure . "
- **Tokenización y tratamiento de las abreviaciones y contracciones**
  Powell said : " we would not be alone ; that is for sure . "

En el primer caso, se emplea la candidata original teniendo en cuenta únicamente los espacios como divisor. En la segunda opción se opta por eliminar todos los signos de puntuación. En el tercer caso, se emplea la herramienta *mteval* que tokeniza todos los signos de puntuación excepto los guiones y los puntos decimales. Leusch añade un cuarto método implementando un tratamiento a las contracciones más comunes en inglés además de mantener todas las palabras en minúsculas.

Por otro lado, se estudian otros factores que pueden variar el resultado de las evaluaciones como la consideración o no de mayúsculas y minúsculas. Leusch encuentra que ignorar las mayúsculas da como resultado puntuaciones más altas.

Se ha optado en este caso por una tokenización de los signos de puntuación, pero teniendo en cuenta los apóstrofos como signo de puntuación. Para ello, se han empleado las siguientes líneas de código:

- Adición espacios delante y detrás de los signos de puntuación.

```
DD_candidate = [ re.sub('(?<! )(?=[.,!?()\'])|(?<=[.,!?()\'])(?! )'
, r' ', DD_candidate[i]) for i in range(n_DD)]
DD_reference = [ re.sub('(?<! )(?=[.,!?()\'])|(?<=[.,!?()\'])(?! )'
, r' ', DD_reference[i]) for i in range(n_DD)]
```

- Cambio apóstrofos curvos por apóstrofos simples.

```
DD_candidate_2 = [re.sub(r'(\')', '\'', DD_candidate[i]) for i in r
ange(n_DD)]
DD_reference_2 = [re.sub(r'(\')', '\'', DD_reference[i]) for i in r
ange(n_DD)]
```

- Tokenización.

```
DD_ref_list_3 = [DD_reference_2[i].split() for i in range(n_DD)]
DD_can_list_3 = [DD_candidate_2[i].split() for i in range(n_DD)]
```

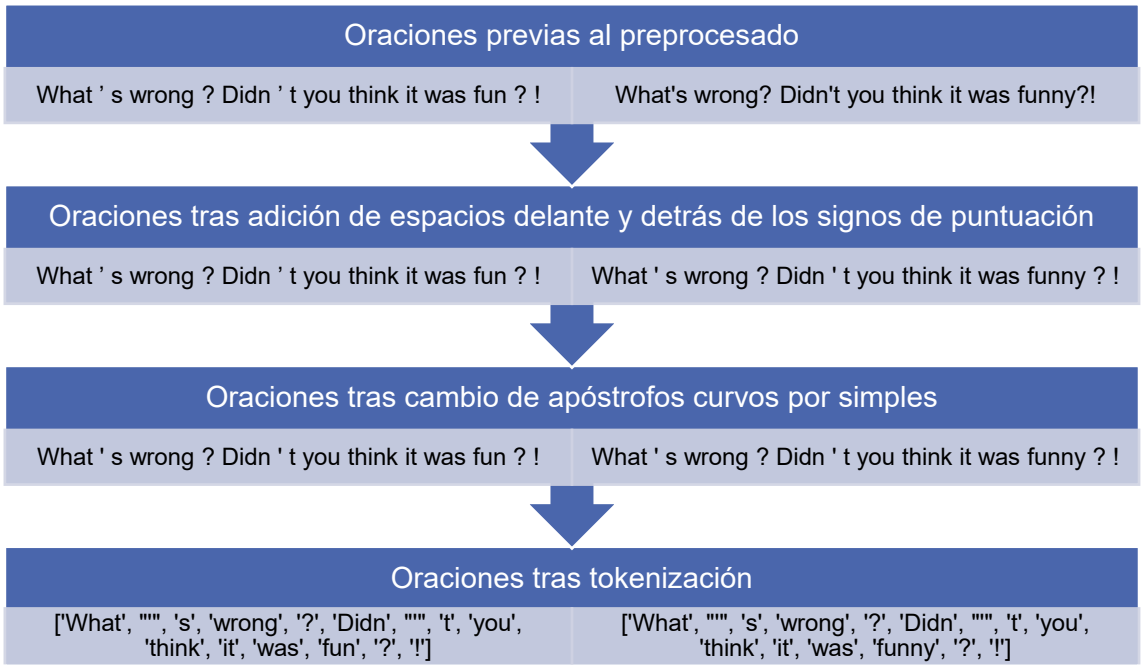



**ILUSTRACIÓN 44. Ejemplo de dos oraciones, original y candidata, en cada paso del proceso de tokenización.**

### 3.1.2 Resultados obtenidos

Una vez realizado el preprocesado de ambas bases de datos, se ha procedido a obtener las valoraciones de la métrica BLEU. La librería necesaria para obtener las valoraciones de BLEU es NLTK y dentro de esta, como se ha indicado anteriormente, la clase *bleu_score*.

Se emplean dos funciones de *bleu_score*:

- Sentence_bleu.

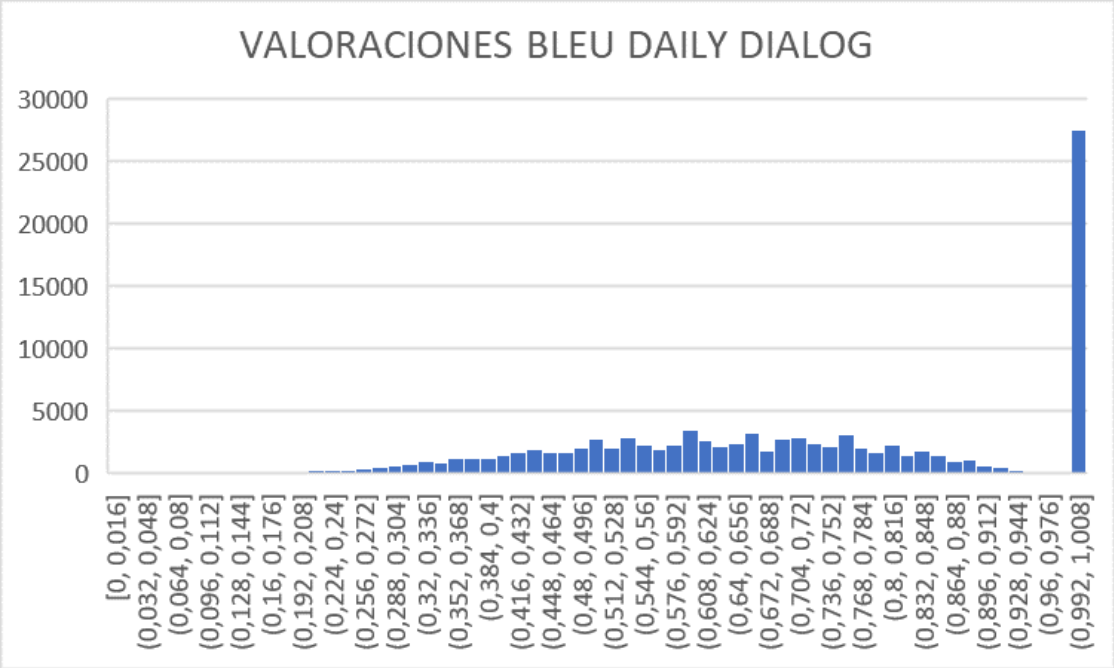



**ILUSTRACIÓN 45. Histograma de las valoraciones de BLEU de DD.**

El resultado del estudio estadístico se muestra en la siguiente tabla:

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,715 |
| **Varianza** | 0,049 |
| **Desviación estándar** | 0,221 |
| **Coeficiente de Variación** | 0,308 |
| **Valor mínimo (Xmín)** | 0 |
| **Cuartil 1** | 0,541 |
| **Mediana (cuartil 2)** | 0,707 |
| **Cuartil 3** | 1 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 1 |

**TABLA 9. Estadística descriptiva de las valoraciones de BLEU para DD.**

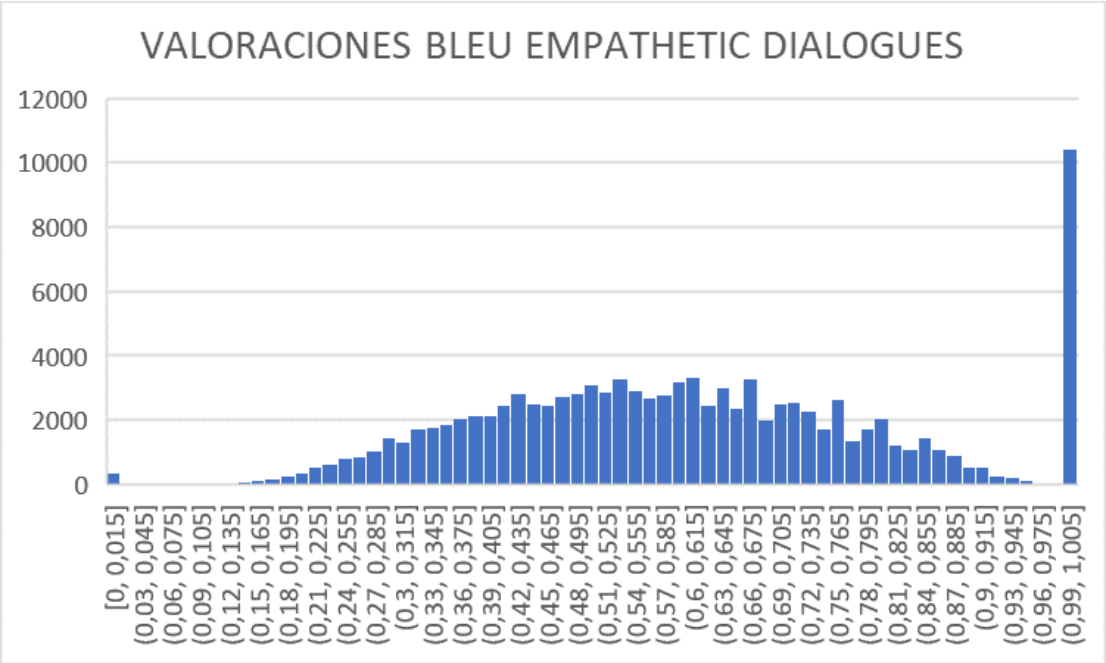



**ILUSTRACIÓN 46. Histograma de las valoraciones de BLEU de ED.**

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,602 |
| **Varianza** | 0,043 |
| **Desviación estándar** | 0,209 |
| **Coeficiente de Variación** | 0,347 |
| **Valor mínimo (Xmín)** | 0 |
| **Cuartil 1** | 0,447 |
| **Mediana (cuartil 2)** | 0,588 |
| **Cuartil 3** | 0,74 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 1 |

**TABLA 10. Estadística descriptiva de las valoraciones de BLEU para ED.**

- Corpus_bleu. En lugar de realizar la media de todas las valoraciones a nivel de oración, *corpus_bleu* suma todos los numeradores y denominadores antes de la división.

  El resultado obtenido para la base de datos de DD es de **0,695**.

  El resultado obtenido para la base de datos de ED es de **0,592**.

  Los resultados de BLEU fluctúan entre el 0 y el 1, pero a efectos prácticos, un resultado de entre 0.6 y 0,7 es considerado muy elevado[30].

[30] https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b#:~:text=Even%20though%20it%20has%20many,the%20best%20you%20can%20achieve.

## 3.2 Obtención valoraciones con modelos de Sentence Transformers

Para solventar el problema que tiene BLEU a la hora de valorar la similitud en la semántica de las oraciones, se ha optado por emplear los modelos de ST. Estos modelos aplican redes siamesas a BERT y son capaces de generar embeddings semánticamente significativos. De esta manera, se pueden trabajar campos previamente inexplorados por BERT como comparación de similitud, que es la que se ha empleado en el desarrollo de este trabajo.

### 3.2.1 Primer método de evaluación. Estudio de un único modelo de Sentence Transformers para la base de datos Daily Dialog

Dentro de ST, existe una gran cantidad de modelos pre entrenados a disposición del usuario. Su utilización es muy sencilla y en el propio sitio web, se ponen a disposición ejemplos de uso para poder implementarlo con los datos del usuario.

Como primer experimento, se optó por comprobar la similitud semántica de las oraciones originales en inglés y de las traducciones al español.

Dentro de los modelos multilingües existentes en ST, se escogieron los siguientes seis para su posterior uso:

- Modelo_0_multi: *paraphrase-multilingual-mpnet-base-v2.* Versión multilingüe de *paraphrase-mpnet-base-v2*. Entrenado con bases de datos paralelas para más de cincuenta idiomas.
- Modelo_1_multi: *paraphrase-multilingual-MiniLM-L12-v2*. Versión multilingüe de *paraphrase -MiniLM-L12-v2*. Entrenado con bases de datos paralelas para más de cincuenta idiomas.
- Modelo_2_multi: *distiluse-base-multilingual-cased-v1*. Versión destilada multilingüe de *multilingual Universal Sentence Encoder*. Soporta 15 idiomas entre los que se encuentra el español y el inglés.
- Modelo_3_multi: *paraphrase-xlm-r-multilingual-v1*. Versión multilingüe de *paraphrase-distilroberta-base-v1*. Entrenado con bases de datos paralelas para más de cincuenta idiomas.
- Modelo_4_multi: *stsb-xlm-r-multilingual*. Obtiene resultados similares a *stsb-bert-base*. Entrenado con bases de datos paralelas para más de cincuenta idiomas.
- Modelo_5_multi: *quora-distilbert-multilingual*. Versión multilingüe de *quora-distilbert-base*. Entrenamiento fino con bases de datos paralelas para más de cincuenta idiomas.

Dada la complicación de estudiar en profundidad cada uno de los modelos, se decidió escoger uno de los modelos para realizar un acercamiento más completo a los resultados obtenidos.

Para escoger el modelo, se puntuaron manualmente 392 intervenciones, empleando un sencillo programa que muestra la intervención origen y la traducida y pide una entrada por teclado de la valoración estimada.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

from google.colab import drive
drive.mount('/content/drive')

%cd /content/drive/My \Drive/Datos/BLEU

DD_reference_dataset = pd.read_csv("DAILYD_translation_en2es.csv")
DD_reference = DD_reference_dataset['SEG']
DD_translation = DD_reference_dataset['translation']
mis_puntuaciones = []

for x in range(len(DD_reference)):
  print(x, DD_reference[x])
  print(DD_translation[x])
  puntuacion = input('Insertar puntuación de la traducción')
  mis_puntuaciones.append(puntuacion)
```

```
df=pd.DataFrame(mis_puntuaciones)
df.to_excel(excel_writer = "manual_scores.xlsx")
```

**ILUSTRACIÓN 47 Script del programa de puntuación manual.**

Los resultados obtenidos, se pueden ver representados en el gráfico siguiente:

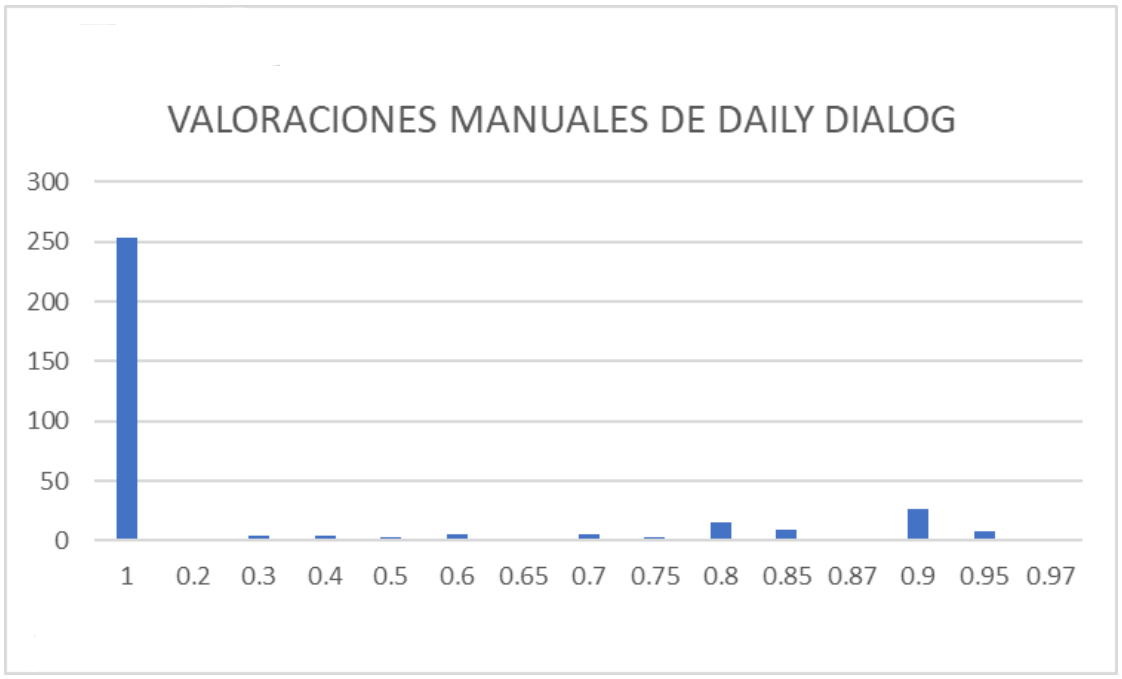



**ILUSTRACIÓN 48. Gráfico de valoraciones de las puntuaciones manuales.**

| Valoración | Cuenta de Valoración |
|---|---|
| **1** | 253 |
| **0,2** | 2 |
| **0,3** | 4 |
| **0,4** | 4 |
| **0,5** | 3 |
| **0,6** | 6 |
| **0,65** | 1 |
| **0,7** | 6 |
| **0,75** | 3 |
| **0,8** | 16 |
| **0,85** | 9 |
| **0,87** | 1 |
| **0,9** | 26 |
| **0,95** | 8 |
| **0,97** | 1 |

**TABLA 11. Tabla con los resultados de las valoraciones manuales.**

La mayor parte de las puntuaciones son elevadas, ya que las traducciones son en general de gran calidad.

Una vez obtenido el vector que representa las puntuaciones obtenidas manualmente se comparó con los vectores de los seis modelos estudiados. Para cuantificar la similitud entre los vectores se emplearon dos métodos:

**SIMILITUD COSENO**

Como se explica previamente, la similitud coseno es un método para obtener la similitud entre los vectores. Se empleó para obtener estos resultados la librería de Python Scipy. En concreto, se hizo uso de la función:

```
scipy.spatial.distance.cosine(u, v, w=None)
```

Donde *u* es el primer vector empleado, *v* el segundo y *w* es el peso de cada uno de los vectores, que por defecto es uno.

El resultado obtenido fue el siguiente:

| MODELO | SIMILITUD COSENO |
|---|---|
| **MODELO_0_MULTI** | 0,9813 |
| **MODELO_1_MULTI** | 0,9803 |
| **MODELO_2_MULTI** | 0,9824 |
| **MODELO_3_MULTI** | 0,9837 |
| **MODELO_4_MULTI** | 0,9803 |
| **MODELO_5_MULTI** | 0,9815 |

**TABLA 12. Valores de similitud coseno entre las valoraciones de los modelos de ST y las valoraciones manuales.**

Como se puede observar, el modelo número tres es el que obtiene el mejor resultado.

**COEFICIENTE DE PEARSON**

Además de la similitud coseno, se optó por emplear un segundo método para valorar la similitud de los vectores. El coeficiente de Pearson se calculó con la función de Numpy:

```
numpy.corrcoef(x, y=None, rowvar=True, dtype=None)
```

Donde *x*, es un vector unidimensional que representa una variable; *y* es un vector adicional con la misma forma que *x*. *rowvar* es un parámetro opcional que se emplea para indicar son las columnas o las filas las que representan variables. *dtype* representa el tipo de dato del resultado, por ejemplo, float. El resultado de esta función es una matriz de coeficientes de las variables.

El resultado fue el siguiente:

| MODELO | COEF PEARSON |
|---|---|
| **MODELO_0_MULTI** | 0,26 |
| **MODELO_1_MULTI** | 0,24 |
| **MODELO_2_MULTI** | 0,27 |
| **MODELO_3_MULTI** | 0,30 |
| **MODELO_4_MULTI** | 0,22 |
| **MODELO_5_MULTI** | 0,14 |

**TABLA 13. Coeficientes Pearson entre los vectores obtenidos con los modelos de ST y las valoraciones manuales.**

En este caso, el modelo tres es también el que ha recibido una puntuación más alta. Ninguno de los modelos obtiene una puntuación especialmente alta. De hecho, solo el modelo tres entra dentro del rango de correlación moderada mientras que los demás modelos presentan una correlación débil.

Escogido el modelo, **Modelo_3_multi:** ***paraphrase-xlm-r-multilingual-v1*** se obtuvieron las valoraciones de todas las intervenciones para dicho modelo con los siguientes resultados.

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,923 |
| **Varianza** | 0,006 |
| **Desviación estándar** | 0,077 |
| **Coeficiente de Variación** | 0,083 |
| **Valor mínimo (Xmín)** | 0 |
| **Cuartil 1** | 0,904 |
| **Mediana (cuartil 2)** | 0,947 |
| **Cuartil 3** | 0,972 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 1 |

**TABLA 14. Estadística descriptiva de las valoraciones del Modelo 3_multi.**

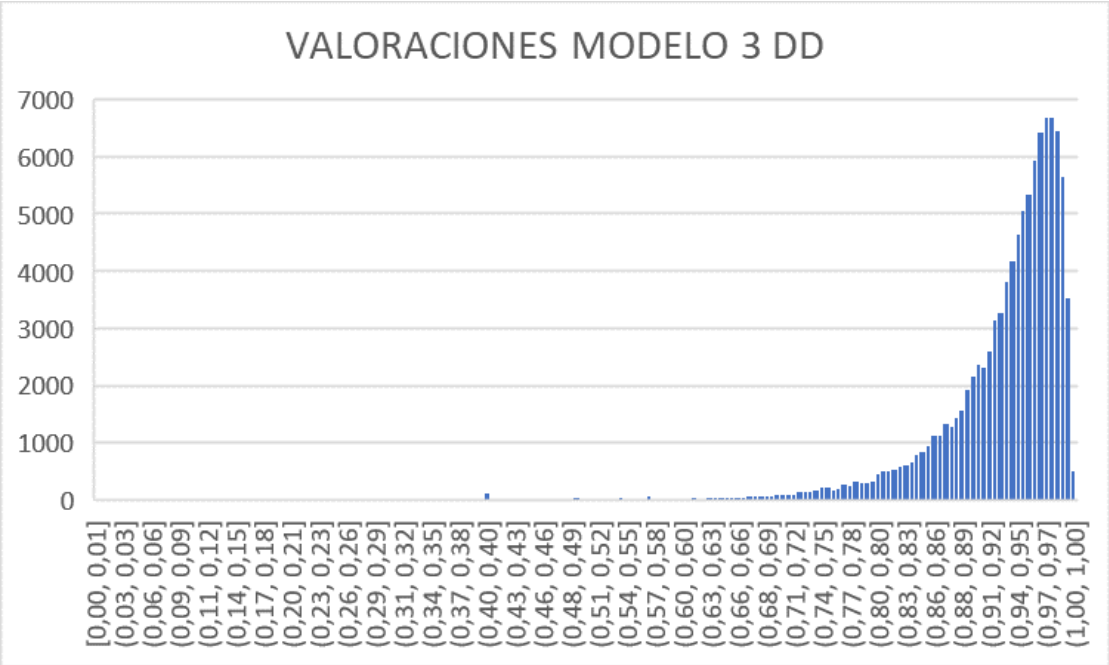



**ILUSTRACIÓN 49. Histograma de los resultados obtenidos con el Modelo 3_multi.**

El primer cuartil, que deja por debajo el 25% de la distribución, se sitúa en 0.904. Esto nos indica que la mayor parte de las respuestas tienen puntuaciones muy elevadas considerando que 1 es la mayor puntuación posible. Para facilitar su estudio, se han dividido las respuestas en grupos. De cada grupo se ha tomado un porcentaje para su revisión manual con el fin de detectar posibles errores recurrentes.

- Grupo A. Puntuaciones $p \leq 0.6$.
- Grupo B. Puntuaciones $0.6 < p < 0.7$.
- Grupo C. Puntuaciones $0.7 < p \leq 0.8$.
- Grupo D. Puntuaciones $0.8 < p \leq 0.9$.
- Grupo E. Puntuaciones $0.9 < p \leq 0.95$.
- Grupo F. Puntuaciones $0.95 < p \leq 1$.

En la siguiente tabla se muestra el número de valoraciones que se han obtenido para cada uno de los grupos de estudio (frecuencia absoluta), el porcentaje que representan las valoraciones de cada grupo frente al total de valoraciones (frecuencia relativa) y la frecuencia relativa acumulada.

| GRUPO | FRECUENCIA ABSOLUTA | FRECUENCIA RELATIVA | FRECUENCIA RELATIVA ACUMULATIVA |
|---|---|---|---|
| **A** | 889 | 0,863% | 0,863% |
| **B** | 1.071 | 1,040% | 1,903% |
| **C** | 3.864 | 3,752% | 5,655% |
| **D** | 18.296 | 17,769% | 23,424% |
| **E** | 29.948 | 29,085% | 52,509% |
| **F** | 48.898 | 47,491% | 100% |

**TABLA 15. Tabla de frecuencias absoluta y relativa de cada grupo de estudio.**

Cabe destacar los errores recurrentes encontrados en los grupos A y B:

- Las intervenciones originales 'Thank you very much', 'Thanks a lot' y 'Thank you so much' traducidas como 'Muchas gracias' obtuvieron puntuaciones de 0.40, 0.49 y 0.38 respectivamente a pesar de ser totalmente correctas.

- Se encontraron intervenciones que habían tenido algún error en el traductor de Azure y devolvían intervenciones del tipo:

    - 'Si estás utilizando un ordenador portátil o una tablet, intenta moverte a otra ubicación e inténtalo de nuevo'
    - 'Hemos detectado un problema'
    - 'Hemos detectado un problema desconocido'
    - 'Si aún así sigues teniendo problemas, problema con la página web'
    - 'Si aún así sigues teniendo problemas, problema desconocido'
    - 'Si aún sigues teniendo problemas, un problema te ayudas a otro'

- Las intervenciones originales *'You are welcome'* traducidas como *'De nada'* obtuvieron puntuaciones de 0.43 a pesar de ser correctas.

En todos los grupos, se encuentran otros errores recurrentes:

- En las intervenciones originales que terminan con el complemento directo *'that'* se traduce como el complementador[31] *'que'* incorrectamente. Se muestra un ejemplo a continuación.

*I can accept that -> Puedo aceptar que (INCORRECTA)*
*Se observa que en la traducción en español falta la oración subordinada sustantiva tras el complementador 'que'.*

*I can accept that -> Puedo aceptarlo/ Lo puedo aceptar (CORRECTA)*
*La traducción correcta emplearía el pronombre complemento directo 'lo'.*

- En las intervenciones originales que comienzan con *'great'* se traducen como *'gran'*.

*Great! I'll just sit here and wait. -> ¡Gran! Me sentaré aquí y esperaré. (INCORRECTA)*
*Se observa que en la traducción en español falta un sustantivo al que complemente el adjetivo gran.*

*Great! I'll just sit here and wait. -> ¡Genial! Me sentaré aquí y esperaré. (CORRECTA)*
*La traducción correcta emplea el sustantivo genial.*

- Intervenciones que presentan en su traducción palabras en inglés que no se han traducido.

[31] https://es.wikipedia.org/wiki/Oraci%C3%B3n_subordinada

*Do you sell stockings? -> Do vendes medias?*
*(INCORRECTA)*

*Do you sell stockings? -> Vendes medias?*
*(CORRECTA)*

- Cambio de registro dentro de una misma intervención.

*So I can say now that you must have enjoyed the opera. Which scene is your favorite? -> Así que puedo decir ahora que usted debe haber disfrutado de la ópera. ¿Qué escena es tu favorita?*

- Nombres propios traducidos al español.

*Mr. Sun -> Sr. Sol*
*Mr. Smith -> Sr. Herrero*
*Mr. Hunter -> Sr. Cazador*

- Uno de los errores más difíciles de detectar y que se encuentra un número considerable de intervenciones se debe a expresiones idiomáticas y frases hechas traducidas literalmente. Se muestra un ejemplo a continuación.

*Oh yeah! I had a blast! I love sweating like a pig with a bunch of pot bellies who all smell bad. Sorry, I'm just no into this health kick. -> ¡Venga, sí! Tuve una explosión! Me encanta sudar como un cerdo con un montón de barrigas de olla que todos huelen mal. Lo siento, simplemente no estoy en la patada de salud.*

*I had a blast -> Tuve una explosión! (INCORRECTA)*
*I had a blast -> ¡Me lo pasé genial! (CORRECTA)*

*bunch of pot bellies -> un montón de barrigas de olla (INCORRECTA)*
*bunch of pot bellies -> un montón de barrigones (CORRECTA)*

*I'm just not into this health kick -> simplemente no estoy en esta patada de salud. (INCORRECTA)*
*I'm just not into this health kick -> simplemente no me va esto de la vida sana(CORRECTA)*

- Intervenciones originales con errores gramaticales que se traducen y dan lugar a intervenciones erróneas en español. Se muestra un ejemplo a continuación.

*What's wrong with that? Cigarette is the thing I go crazy for. -> ¿Qué tiene de malo? Cigarrillo es lo que me vuelvo loco por.*

*Subrayado y en rojo se destaca el error gramatical en la intervención original. En lugar de 'cigarette is' debería decir 'cigarettes are'.*

#### 3.2.1.1 Procesado de la base de datos

Una vez detectados los errores más destacables en la base de datos se procedió a procesar los resultados. Por un lado, se cambió la puntuación de aquellas intervenciones que contenían un *'muchas gracias'* o un *'de nada'* en alguna de sus oraciones. Por otro lado, se eliminaron las puntuaciones de aquellas intervenciones que contenían mensajes de error. Para ello se empleó el código que se muestra a continuación.

```
datos=[]
n_0 = n_1 = n_2 =0
for x in range(len(DD_reference)-1):
  if Puntuaciones[x] >= 0 and Puntuaciones[x]<=1:
    if DD_translation[x].find("Hemos detectado un problema ") != -
1 or DD_translation[x].find("Si estás utilizando un ordenador") != -
1 or DD_translation[x].find("sigues teniendo problemas") != -
1 or DD_translation[x].find("Hemos detectado un error desconocido") != -1:
      print(x, DD_reference[x])
      print(DD_translation[x])
      print(Puntuaciones[x])
      n_0 = n_0+1
  if Puntuaciones[x] >=0 and Puntuaciones[x]<=0.7:
    if DD_translation[x].find("Muchas gracias") != -1:
      print(x, DD_reference[x])
      print(DD_translation[x])
      print(Puntuaciones[x])
      n_1 = n_1+1
      print("Nueva puntuación")
      score=input()
      vector=[score,x]
      datos.append(vector)
    if DD_translation[x].find("De nada") != -1:
      print(x, DD_reference[x])
      print(DD_translation[x])
      print(Puntuaciones[x])
      n_2 = n_2+1
      print("Nueva puntuación")
      score=input()
      vector=[score,x]
      datos.append(vector)
  else:
    print(x, DD_reference[x])
    print(DD_translation[x])
    print(Puntuaciones[x])
    vector=[Puntuaciones[x],x]
    datos.append(vector)
print(n_0, n_1 ,n_2)
```

Donde *Puntuaciones* es el vector de valoraciones del modelo, *DD_reference* es el conjunto de intervenciones de referencia en inglés y *DD_translation* es el conjunto de intervenciones traducidas al español.

Se empleó la siguiente función de Python:

*str.find(sub, start, end)*

Donde *sub* es la subcadena que se desea buscar, *start* es la posición donde debe buscarse sub dentro de la cadena y *end* es la posición donde debe buscarse el sufijo dentro de la cadena. Esta función devuelve el índice más bajo donde se ha encontrado la subcadena. Si no encuentra la subcadena devuelve un -1.

Se encontraron 258 intervenciones con mensajes de error, 289 intervenciones traducidas como 'Muchas gracias' con una falsa puntuación negativa y 156 intervenciones traducidas como 'De nada' con una falsa puntuación negativa.

Tras reasignar puntuaciones más elevadas a los falsos negativos y eliminar las intervenciones con errores, se observó que el 98.53% de las puntuaciones se encontraban en el intervalo de 0.7 a 1. Por lo que se consideraron valores sesgados y se reetiquetaron las puntuaciones obtenidas, obteniendo un total de 101499 puntuaciones.

Para mayor claridad, también se muestra a continuación una segunda gráfica con las puntuaciones delimitadas inferiormente por 0.8. De esta manera, se mantienen 97853 puntuaciones, lo que corresponde a un 95.03% de las puntuaciones iniciales.

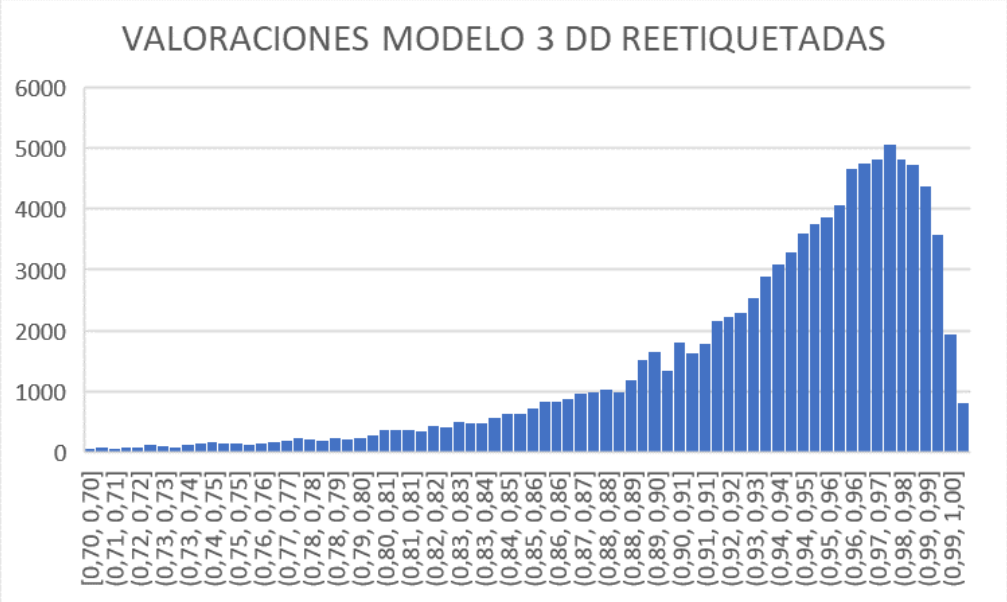



**ILUSTRACIÓN 50. Histograma de las valoraciones de Modelo_3_multi tras el reetiquetado de 0,7 a 1.**

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,932 |
| **Varianza** | 0,003 |
| **Desviación estándar** | 0,056 |
| **Coeficiente de Variación** | 0,060 |
| **Valor mínimo (Xmín)** | 0,7 |
| **Cuartil 1** | 0,908 |
| **Mediana (cuartil 2)** | 0,948 |
| **Cuartil 3** | 0,973 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 0,3 |

**TABLA 16 Descripción estadística de las valoraciones de Modelo_3_multi tras el reetiquetado entre 0.7 y 1.**

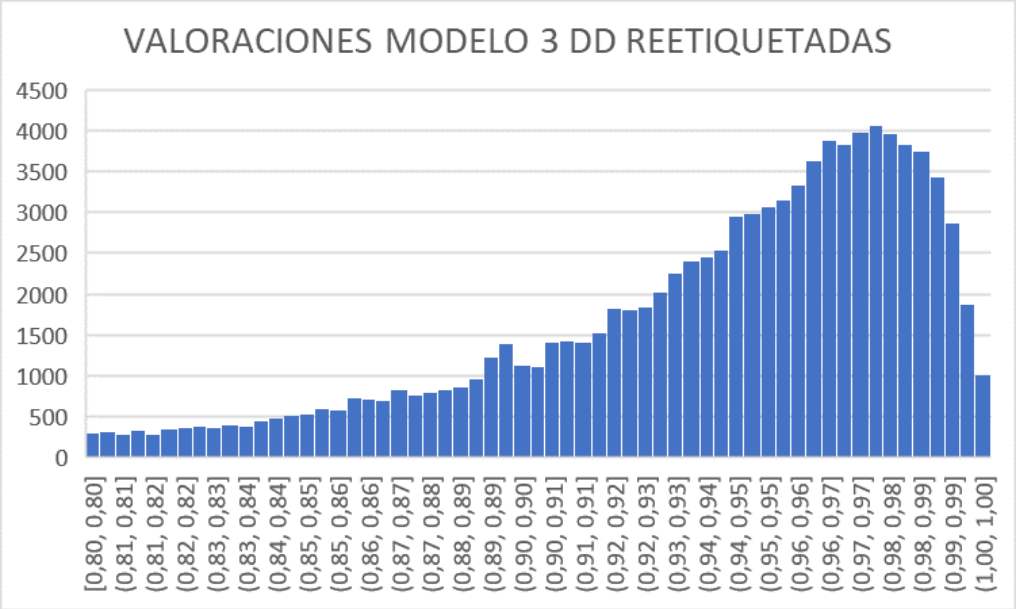



**ILUSTRACIÓN 51. Histograma de las valoraciones del Modelo_3_multi tras el reetiquetado de 0,8 a 1.**

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,939 |
| **Varianza** | 0,002 |
| **Desviación estándar** | 0,045 |
| **Coeficiente de Variación** | 0,048 |
| **Valor mínimo (Xmín)** | 0,8 |
| **Cuartil 1** | 0,915 |
| **Mediana (cuartil 2)** | 0,951 |
| **Cuartil 3** | 0,974 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 0,2 |

**TABLA 17 Descripción estadística de las valoraciones del Modelo_3_multi tras el reetiquetado entre 0.8 y 1.**

### 3.2.2 Primer método de evaluación. Estudio de un único modelo de Sentence Transformers para la base de datos de Empathetic Dialogues

Una vez realizados los primeros dos experimentos con la base de datos de DD, se procedió a estudiar la base de datos de ED. Se empleó el **Modelo_3: *paraphrase-xlm-r-multilingual-v1*** para realizar estudiar esta base de datos porque es la que obtuvo mejores resultados en cuanto a la similitud con las valoraciones manuales. Se encuentra la información sobre el proceso de elección en el apartado 3.2.1 de este documento.

Los resultados obtenidos para la base de datos de ED fueron los siguientes:

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,912 |
| **Varianza** | 0,005 |
| **Desviación estándar** | 0,074 |
| **Coeficiente de Variación** | 0,081 |
| **Valor mínimo (Xmín)** | 0 |
| **Cuartil 1** | 0,886 |
| **Mediana (cuartil 2)** | 0,932 |
| **Cuartil 3** | 0,960 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 1 |

**TABLA 18. Estadística descriptiva de las valoraciones del Modelo_3_multi.**

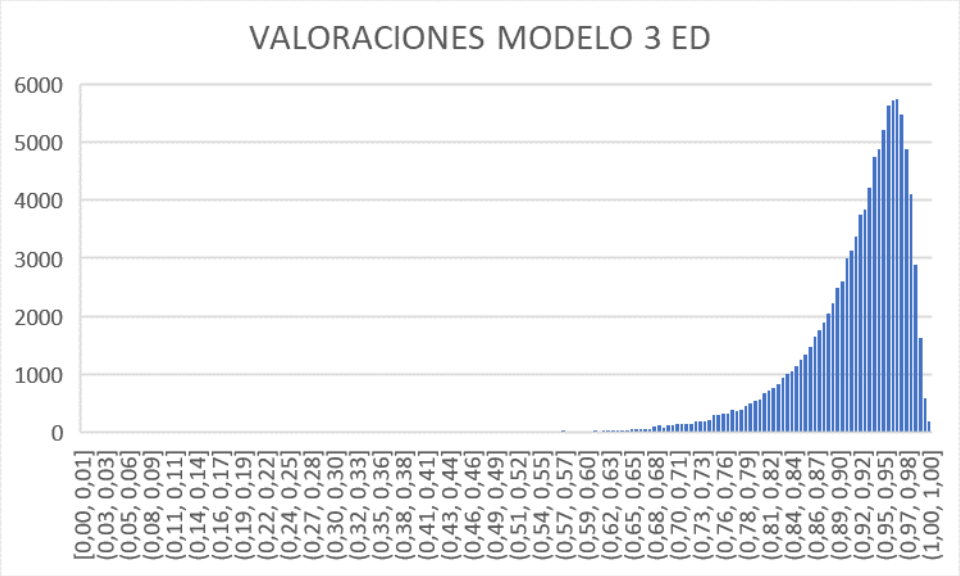



**ILUSTRACIÓN 52. Histograma de los resultados obtenidos con el Modelo_3_multi.**

Como se puede observar en el histograma, la mayor parte de los resultados se encuentran desplazados hacia la parte derecha del eje horizontal, al igual que se observó con la base de datos de DD.

Para este caso, también se dividieron las valoraciones en grupos de estudio. Los grupos en este caso fueron los siguientes:

- Grupo A. Puntuaciones $p \leq 0.6$.
- Grupo B. Puntuaciones $0.6 < p < 0.7$.
- Grupo C. Puntuaciones $0.7 < p \leq 0.8$.
- Grupo D. Puntuaciones $0.8 < p \leq 0.9$.
- Grupo E. Puntuaciones $0.9 < p \leq 0.95$.
- Grupo F. Puntuaciones $0.95 < p \leq 1$.

| GRUPO | FRECUENCIA ABSOLUTA | FRECUENCIA RELATIVA | FRECUENCIA RELATIVA ACUMULATIVA |
|---|---|---|---|
| **A** | 595 | 0,554% | 0,554% |
| **B** | 1.279 | 1,193% | 1,747% |
| **C** | 5.621 | 5,242 % | 6,989% |
| **D** | 25.709 | 23,977% | 30,966% |
| **E** | 36.690 | 34,222% | 65,188% |
| **F** | 37.324 | 34,812% | 100% |

**TABLA 19. Tabla de frecuencias absoluta y relativa de cada grupo de estudio.**

Se analizaron intervenciones aleatorias de cada grupo para comprobar los errores, falsos positivos o falsos negativos.

En el Grupo A, se encontraron los siguientes errores:

- Mensajes de error del traductor de Azure. Es estos casos, se traduce parte de la intervención original o simplemente se encuentra el mensaje de error. Los errores muestran los siguientes mensajes:
  - 'Si aún así sigues teniendo problemas, visita la página de ayuda de Hasta el final.'
  - 'Hemos detectado un problema desconocido.'
  - 'Si estás utilizando un ordenador portátil o una tablet, intenta moverte a otra ubicación e inténtalo de nuevo.'
  - 'Si aún así sigues teniendo problemas, problema con la página de ayuda de Ayuda de Copa('
  - 'Si aún así sigues teniendo problemas, intenta tu página de ayuda de nuevo.'

- Errores ortográficos en las oraciones originales como:

  *'thats cool'* → '*eso está estupendo'*

  *'thats cool' (INCORRECTA)* → *'That's cool' (CORRECTA)*

- Oraciones originales sin signos de interrogación en las preguntas. En la oración traducida se añade el signo de interrogación al comienzo de la pregunta y en algunos casos no traduce la oración por completo.

‘*Who is the quarterback now’* → *‘¿Quién es el mariscal de campo ahora’ (INCORRECTA)*

‘*Who is the quarterback now?’* → *‘¿Quién es el mariscal de campo ahora?’ (CORRECTA)*

- Traducciones incompletas. Algunas de las intervenciones no se han traducido completamente.

*‘that sucks’* → ‘*Es una’ (INCORRECTA*)

‘*that sucks’* → ‘*eso apesta’(CORRECTA)*

- Traducciones incorrectas.

‘*I tought him to sit for a treat, its so cute*.’ → ‘*Lo resistente a sentarse a un regalo, es tan lindo.’ (INCORRECTA)*

*I tought him to sit for a treat, its so cute*.’ → ‘*Le enseñé a sentarse por una chuche, es tan lindo.’ (CORRECTA)*

- Se encontraron intervenciones que traducían ‘Thank you very much’ como ‘Muchas gracias’ como falsos negativos, con puntuaciones de 0,51.

En el grupo B y C se encontraron los mismos errores que en el Grupo A, pero en mucha menor medida. Las oraciones eran en su gran mayoría correctas. Se encontraron algunos ejemplos de oraciones traducidas incorrectamente por falta de contexto, siendo imposible el trabajo de desambiguación:

‘*Did you get a new trim?’* → ‘*¿Tienes un nuevo adorno?’ (INCORRECTA)*

*‘Did you get a new trim?’* → ‘*Te has cortado el pelo?’ (CORRECTA)*

A partir del Grupo D (es decir, las puntuaciones mayores a 0,8), se encontraron algunos errores de traducción, pero en un porcentaje proporcionalmente menor según aumentaban las puntuaciones.

#### 3.2.2.1 Procesado de la base de datos

Una vez analizadas las intervenciones y las puntuaciones obtenidas se realizó el procesado de aquellos falsos negativos y se eliminaron las intervenciones que presentaban un mensaje de error. Se hizo de la misma manera que se describe en el apartado 3.2.1.1 de este documento.

En este caso, se encontraron 294 intervenciones con un mensaje de error, 11 intervenciones traducidas como ‘Muchas gracias’ con una falsa puntuación negativa, 4 intervenciones traducidas como ‘De nada’ con una falsa puntuación negativa.

Procesada la base de datos, se observó que el 98,253% de las intervenciones recibieron una puntuación dentro del intervalo [0,7-1], por lo que se acotó dentro de este intervalo, para representarlo en más claramente en un histograma. Se muestra a continuación el resultado obtenido tras el procesado y el acotado.

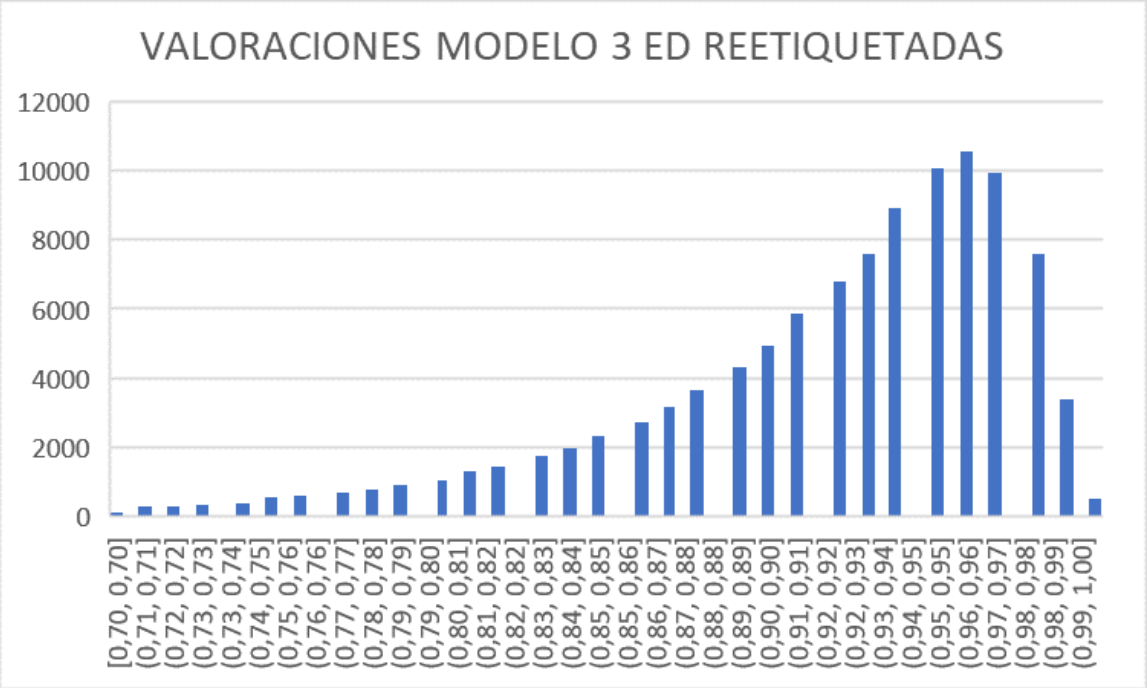



**ILUSTRACIÓN 53. Histograma de las valoraciones del Modelo_3_multi tras el reetiquetado de 0,7 a 1.**

| VARIABLE | VALOR |
|---|---|
| **Media** | 0,917 |
| **Varianza** | 0,003 |
| **Desviación estándar** | 0,058 |
| **Coeficiente de Variación** | 0,063 |
| **Valor mínimo (Xmín)** | 0 |
| **Cuartil 1** | 0,89 |
| **Mediana (cuartil 2)** | 0,93 |
| **Cuartil 3** | 0,96 |
| **Valor máximo (Xmáx)** | 1 |
| **Rango (Xmáx-Xmín)** | 1 |

**TABLA 20. Descripción estadística de las valoraciones del Modelo_3_multi tras el reetiquetado entre 0.7 y 1.**

### 3.2.3 Segundo método de evaluación. Regresión lineal entrenada con modelos de Sentence Transformers para la base de datos de Daily Dialog

Tras estudiar el Modelo_3_multi de ST, se decidió entrenar un modelo de regresión lineal que conectara las valoraciones de ST con las valoraciones de BLEU, al tratarse la última de la métrica de base. El objetivo era encontrar un modelo que obtuviera una evaluación basada en BLEU y en las valoraciones de modelos monolingües sin la necesidad de emplear intervenciones candidatas en inglés. Para ello, se emplearon entradas de modelos multilingües de comparación de paráfrasis y se entrenaron con las valoraciones de un modelo monolingüe.

Para ello, se decidió que se escogerían dos de los modelos multilingües de ST como variables de entrada de la LR y un modelo monolingüe de ST como salida de entrenamiento.

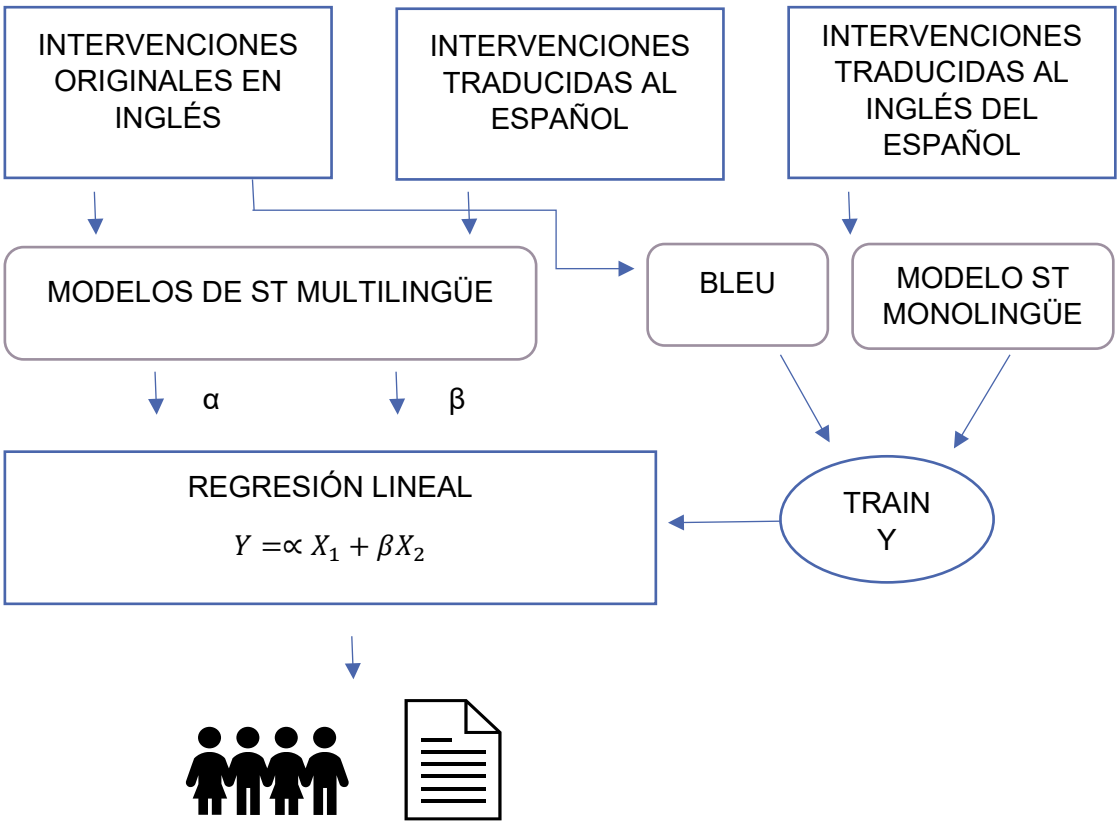



**ILUSTRACIÓN 54. Esquema para la obtención de una LR entrenada.**

Para escoger el modelo monolingüe de entrenamiento que más se asemejara a las valoraciones de BLEU, se procedió de la misma manera que para escoger el modelo multilingüe en el primer experimento.

Para escoger el modelo monolingüe se limitó el número de modelos a cinco de entre los disponibles en la web de ST. Estos modelos se muestran a continuación:

- Modelo_0_mono. *paraphrase-mpnet-base-v2*
- Modelo_1_mono. *paraphrase-TinyBERT-L6-v2*
- Modelo_2_mono. *paraphrase-distilroberta-base-v2*
- Modelo_3_mono. *paraphrase-MiniLM-L12-v2*
- Modelo_4_mono. *paraphrase-MiniLM-L6-v2*

**SIMILITUD COSENO**

De la misma manera que en el anterior experimento, se empleó para obtener estos resultados la función *spatial.distance.cosine() de* la librería de Python Scipy. El vector *u* en este caso es el vector de cada uno de los modelos monolingües de ST y *v* es el vector de valoraciones de BLEU.

El resultado obtenido fue el siguiente:

| MODELO | SIMILITUD COSENO |
|---|---|
| **MODELO_0_MONO** | 0,9653 |
| **MODELO_1_MONO** | 0,9668 |
| **MODELO_2_MONO** | 0,9663 |
| **MODELO_3_MONO** | 0,9665 |
| **MODELO_4_MONO** | 0,9666 |

**TABLA 21. Valores de similitud coseno entre los modelos de ST y las valoraciones de BLEU.**

Como se puede observar, Modelo_1_mono es el que obtiene el mejor resultado.

**COEFICIENTE DE PEARSON**

Como anteriormente, el coeficiente de Pearson se calculó con la función de Numpy *corrcoef():*

El resultado fue el siguiente:

| MODELO | COEF PEARSON |
|---|---|
| **MODELO_0_MONO** | 0,497 |
| **MODELO_1_MONO** | 0,528 |
| **MODELO_2_MONO** | 0,518 |
| **MODELO_3_MONO** | 0,531 |
| **MODELO_4_MONO** | 0,526 |

**TABLA 22. Coeficientes de Pearson entre los vectores obtenidos con los modelos de ST y las valoraciones manuales.**

El mejor resultado es el de Modelo_3_mono.

Se escogió el modelo **Modelo_1_mono: *paraphrase-TinyBERT-L6-v2***, ya que obtuvo el resultado más alto en la similitud coseno y el segundo en el coeficiente de Pearson.

#### 3.2.3.1 Entrenamiento del modelo con dos entradas

Los modelos multilingües escogidos fueron el **Modelo_2_multi: *distiluse-base-multilingual-cased-v1*** y el **Modelo_3_multi: *paraphrase-xlm-r-multilingual-v1*** porque fueron los que más similitud con las valoraciones manuales obtuvieron en el experimento anterior. Se les nombrará Modelo_2_ing_esp y Modelo_3_ing_esp para diferenciarlos del modelo monolingüe, al que se llamará Modelo_1_ing_ing.

Lo primero que se hizo, fue visualizar los datos en forma de diagramas de dispersión, gracias a la función pairplot de la librería Seaborn de Python:

```
sns.pairplot(raw_data, vars=["modelo_1_ing_ing","modelo_2_ing_esp","mo
delo_3_ing_esp"])
```

Obteniendo los siguientes resultados:

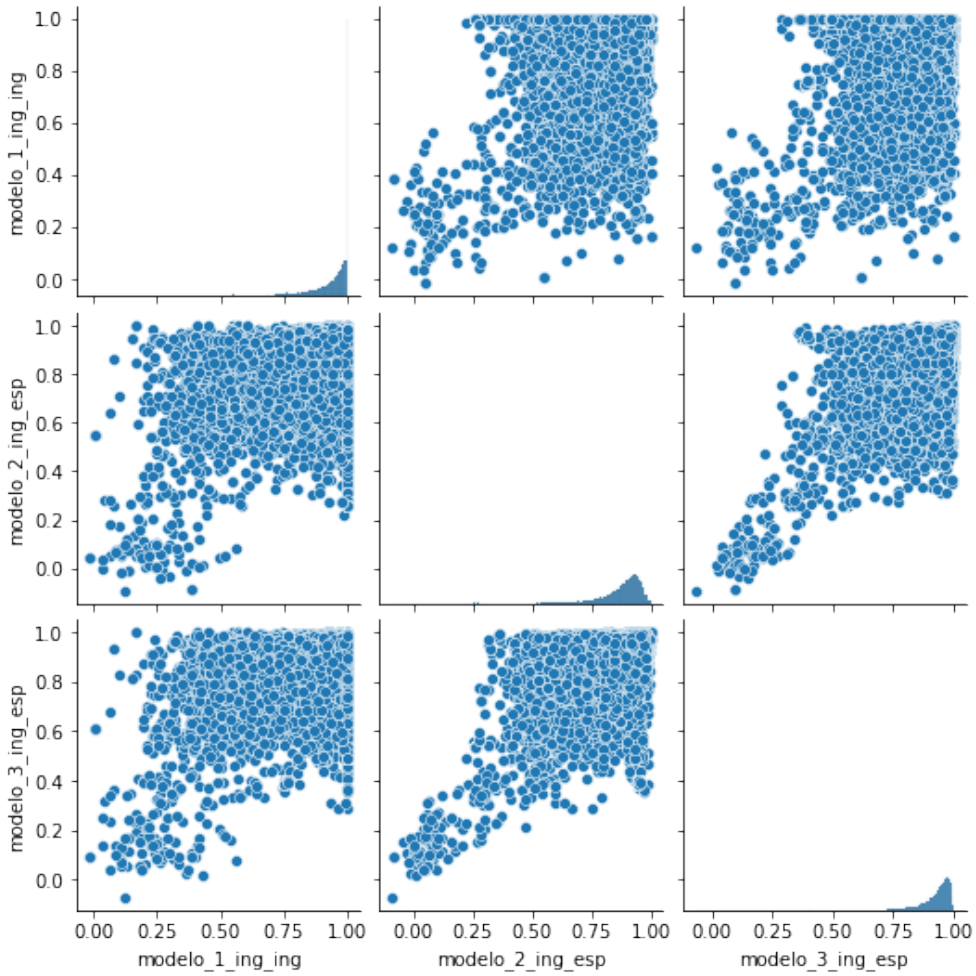



**ILUSTRACIÓN 55. Diagramas de dispersión entre los diferentes modelos.**

Se observó cierta tendencia a la linealidad, especialmente entre los modelos Modelo_2_ing_esp y Modelo_3_ing_esp.

Para entrenar el modelo, se empleó la librería Sklearn de Python.

Las líneas de código empleadas se muestran a continuación:

```
raw_data.columns
y = raw_data['modelo_1_ing_ing']
x = raw_data[['modelo_2_ing_esp','modelo_3_ing_esp']]
```

Se asignan las valoraciones de Modelo_2_ing_esp y Modelo_3_ing_esp a la variable *x* y las de Modelo_1_ing_ing a la variable *y.*

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size
 = 0.3)
```

Mediante la función *train_test_split*, se divide la base de datos. Se puede escoger el porcentaje empleado para entrenamiento y para test. En este caso, se ha determinado que hay un 30% de los datos para el test.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train, y_train)
print(model.coef_)
```

Mediante La función *LinearRegression()* se escoge el modelo que se desea emplear de los disponibles en la librería Sklearn, en este caso una regresión lineal. La función *fit* se utiliza para entrenar el modelo con los datos de entrenamiento. La función *print(model.coef_)* muestra por pantalla el resultado del entrenamiento, es decir, los coeficientes de la regresión lineal.

El resultado obtenido en este caso fue: `[0.16543127 0.5831466]`

```
predictions = model.predict(x_test)
plt.scatter(y_test, predictions)
```

La función *model.predict* permite hacer predicciones con el modelo obtenido. Para ello empleamos las entradas que se habían reservado para el test previamente. Para comparar los valores reales y las predicciones se emplea la función plt.scatter, que nos muestra un diagrama de dispersión.

El gráfico que obtenido fue:

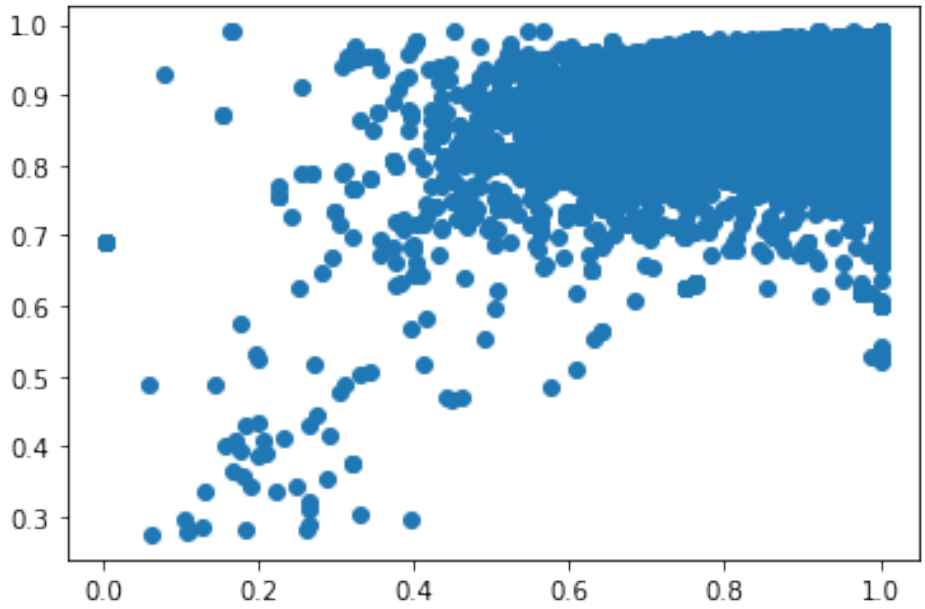



**ILUSTRACIÓN 56. Diagrama de dispersión del modelo de LR entrenado.**

Este resultado no presenta una gran linealidad. El resultado ideal sería una diagonal recta, lo cual indicaría que el modelo realiza una predicción perfecta.

```
plt.hist(y_test - predictions)
```

Además, se puede realizar un gráfico de residuos para ver la diferencia entre las predicciones y las salidas reales. Para ello, se emplea la función *plt.hist(y_test – predictions)*, con el que se obtuvo el siguiente resultado:

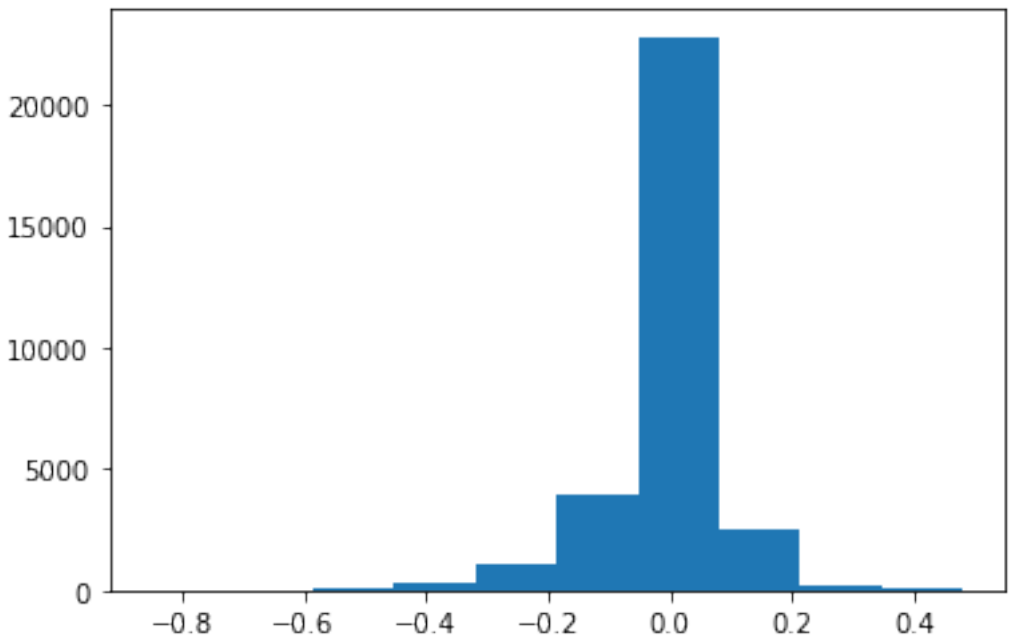



**ILUSTRACIÓN 57. Gráfica de residuos de las predicciones del modelo LR entrenado y las salidas de test.**

Como se puede observar, sigue una distribución similar a una normal, lo cual nos indica que el modelo ha obtenido un resultado aceptable.

```
from sklearn.metrics import r2_score
r2_score(y_test,predictions)
```

Por último, se emplea la función r2_score para obtener el coeficiente de determinación. El coeficiente determina la calidad del modelo para replicar los resultados, y la proporción de la variación de los resultados que puede explicarse por el modelo.

El resultado obtenido fue `0.266`. El modelo obtenido "explica" solo en un 26,6% a la variable real.

#### 3.2.3.2 Entrenamiento del modelo con seis entradas

Ya que el resultado obtenido no fue totalmente satisfactorio, se procedió a aumentar el número de entradas en el modelo, empleando las valoraciones de los seis modelos multilingües en lugar de utilizar únicamente una pareja de modelos.

Los modelos multilingües empleados, fueron los siguientes: Modelo_0: *paraphrase-multilingual-mpnet-base-v2,* Modelo_1: *paraphrase-multilingual-MiniLM-L12-v2,* Modelo_2: *distiluse-base-multilingual-cased-v1*, Modelo_3: *paraphrase-xlm-r-multilingual-v1,* Modelo_4: *stsb-xlm-r-multilingual,* Modelo_5: *quora-distilbert-multilingual.* Y el modelo monolingüe fue el mismo que se empleó anteriormente. Modelo_1_mono.

Se emplearon las mismas llamadas que anteriormente para entrenar al modelo. Se obtuvo el siguiente diagrama de dispersión:

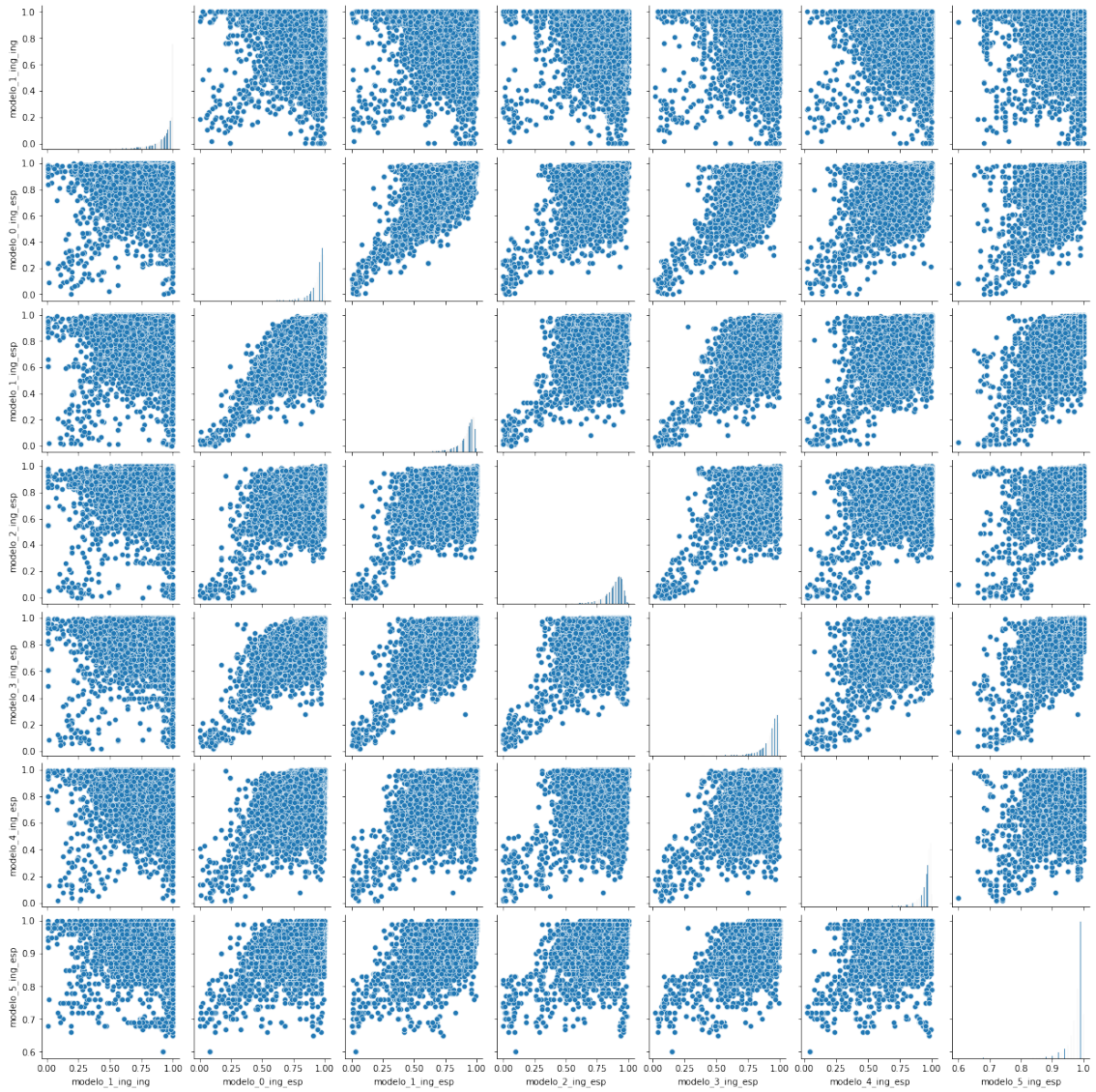



**ILUSTRACIÓN 58. Diagramas de dispersión entre los diferentes modelos.**

Se dividieron los datos en aquellos destinados al entrenamiento, un 70%, y aquellos destinados al test posterior, un 30%. Una vez dividida la base de datos, se procedió a entrenar el modelo con la función *fit()* de la librería Sklearn.

El resultado obtenido fue el siguiente:

```
[0.21823429  0.01476081  0.01544197  0.13443293 -0.00339057 -
0.09809735].
```

Una vez obtenidos los coeficientes, se procedió a probar el modelo con los datos de test. Se obtuvo el diagrama de dispersión, con la siguiente forma:

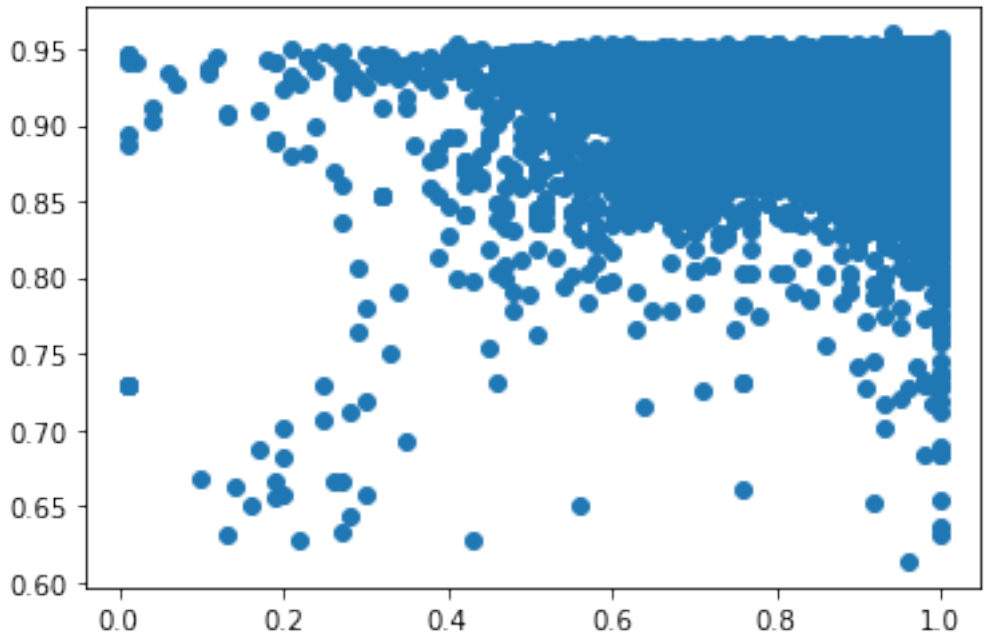



**ILUSTRACIÓN 59. Diagrama de dispersión del modelo de LR entrenado.**

El modelo de dispersión no presenta gran linealidad. Cabe destacar que la gran mayoría de los resultados se concentra en los valores más elevados, lo cual resta claridad al diagrama.

La gráfica de residuos obtenida fue la siguientes:

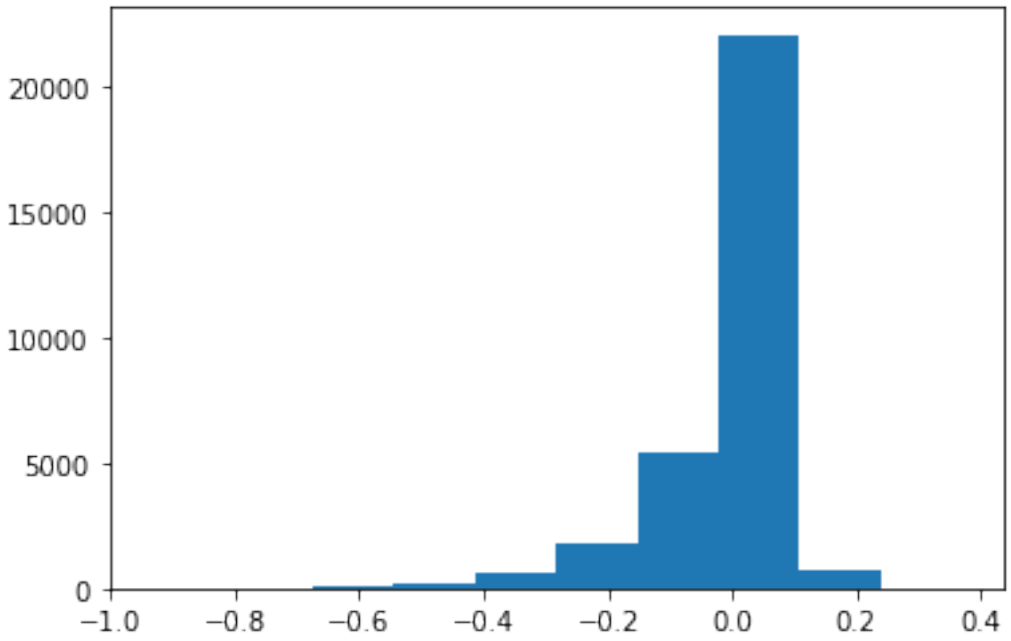



**ILUSTRACIÓN 60. Gráfica de residuos de las predicciones del modelo LR entrenado y las salidas de test.**

Sigue una distribución similar a la normal una vez más con las respuestas concentradas en el valor 0, lo cual es un signo positivo.

El resultado del coeficiente de determinación fue `0.257`, lo cual indica que se explica en un 25,7% a la variable real.

#### 3.2.3.3 Evaluación con test manual

Sin embargo, teniendo en cuenta la distribución del diagrama de residuos, que se concentra alrededor del 0, se optó por realizar un test manual.

Se empleó el vector de salida de test (*y_test*) y el vector de las predicciones del modelo (*predictions*) para valorar manualmente el modelo.

Se emplearon las siguientes líneas de código:

```
n_correcta=0
n_erronea=0
y=y_test.to_list()

for x in range(len(y)):
  resultados.append(abs(predictions[x]- y[x]))
  if abs(predictions[x] - y[x]) < 0.07:
    n_correcta=n_correcta+1
  else :
    n_erronea=n_erronea+1

print(n_correcta/(n_correcta+n_erronea)*100)
```

Donde se emplearon con contabilizadores, *n_correcta* (para aquellos casos en los que se cumple la condición) y *n_erronea* (para aquellos casos que no cumplen la condición).

Se empleó un bucle para evaluar la distancia entre la predicción y el valor real en valor absoluto. Se probaron diferentes distancias y se evaluó el porcentaje de casos sobre el total en el que se había cumplido la condición impuesta. Se muestran en una tabla a continuación:

| Desviación típica relativa | Porcentaje de acierto |
|---|---|
| **0,02** | 14,25% |
| **0,03** | 22,24% |
| **0,04** | 31,14% |
| **0,05** | 41,96% |
| **0,06** | 59,23% |
| **0,07** | 70,08% |
| **0,08** | 77% |

**TABLA 23. Representación de los porcentajes de acierto según la desviación típica relativa entre test y predicción de LR de 6 entradas.**

Estos resultados resultaron satisfactorios, por lo que se calculó el intervalo de confianza de los residuos.

Se emplearon las siguientes líneas de código:

```
import numpy as np
import scipy.stats

def mean_confidence_interval(data, confidence=0.95):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    return m, m-h, m+h

print(mean_confidence_interval(resultados, confidence=0.95))
```

Obteniendo el siguiente resultado por pantalla: `(0.07108755448101438, 0.07027661065462054, 0.07189849830740822)`

La función *mean_confidence_interval* devuelve tres valores:

- m. La media muestral.
- m-h. El extremo inferior del intervalo.
- m+h. El extremo superior del intervalo.

El intervalo de confianza de 95% del vector de residuos por tanto resulta: (0,070;0,071)

Al comprobar que el test manual valoró correctamente el modelo entrenado, se procedió a realizar el mismo test sobre el modelo de dos entradas obteniendo los siguientes resultados:

| Desviación típica relativa | Porcentaje de acierto |
|---|---|
| **0,02** | 22,61% |
| **0,03** | 37,76% |
| **0,04** | 51,16% |
| **0,05** | 61,29% |
| **0,06** | 68,45% |
| **0,07** | 74% |
| **0,08** | 78,28% |

**TABLA 24. Representación de los porcentajes de acierto según la desviación típica relativa entre test y predicción de LR de 2 entradas.**

El intervalo de confianza de 95% del vector de residuos resultó: (0,059;0,060).

# Capítulo 4. Resultados y discusión

En este capítulo, se procederá a analizar los resultados obtenidos por cada uno de los métodos de evaluación en las diferentes bases de datos. Se recogerán los beneficios y los problemas que se encuentran en cada método de evaluación gracias a los análisis estadísticos, las gráficas y otras representaciones que se han obtenido a lo largo del desarrollo del proyecto.

## 4.1 Análisis de los resultados

### 4.1.1 Análisis de los resultados de la métrica BLEU

Se recogerán en una tabla las características básicas de ambas bases de datos para tenerlo en consideración en su análisis y comparación posteriores.

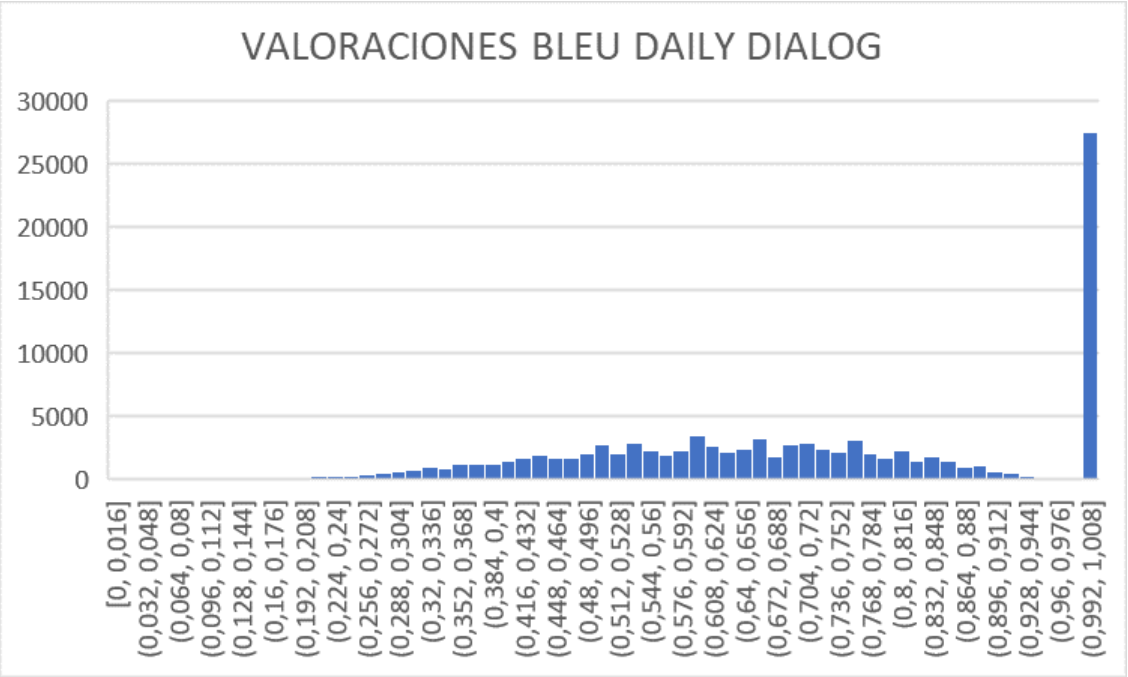
| Característica | DD | ED |
|---|---|---|
| **Número total de diálogos** | 13.118 | 24.850 |
| **Media de turnos de palabra por diálogo** | 7,9 | 4,31 |
| **Media de palabras por intervención** | 14,6 | 15,2 |
| **Total de intervenciones** | 102.968 | 107.220 |

**TABLA 25. Características básicas de DD y ED.**

Como se puede observar ambas bases de datos son similares. La mayor diferencia reside en que DD tiene un número menor de diálogos, pero con más intervenciones por diálogo, un 83,29% más que ED.

La primera evaluación se realizó con la métrica de evaluación BLEU. Se empleó esta métrica porque se trata de la baseline (en español métrica de referencia, baseline es un término inglés empleado dentro del campo de la investigación) dentro del campo del NLP. Se tomó la métrica BLEU como un preanálisis para comprobar que las traducciones eran correctas a un nivel general, con vistas a realizar una evaluación más exhaustiva, en caso de que los resultados de BLEU fueran aceptables, con modelos que tuvieran en consideración la semántica de la oración.

Los resultados obtenidos por BLEU fueron los siguientes:



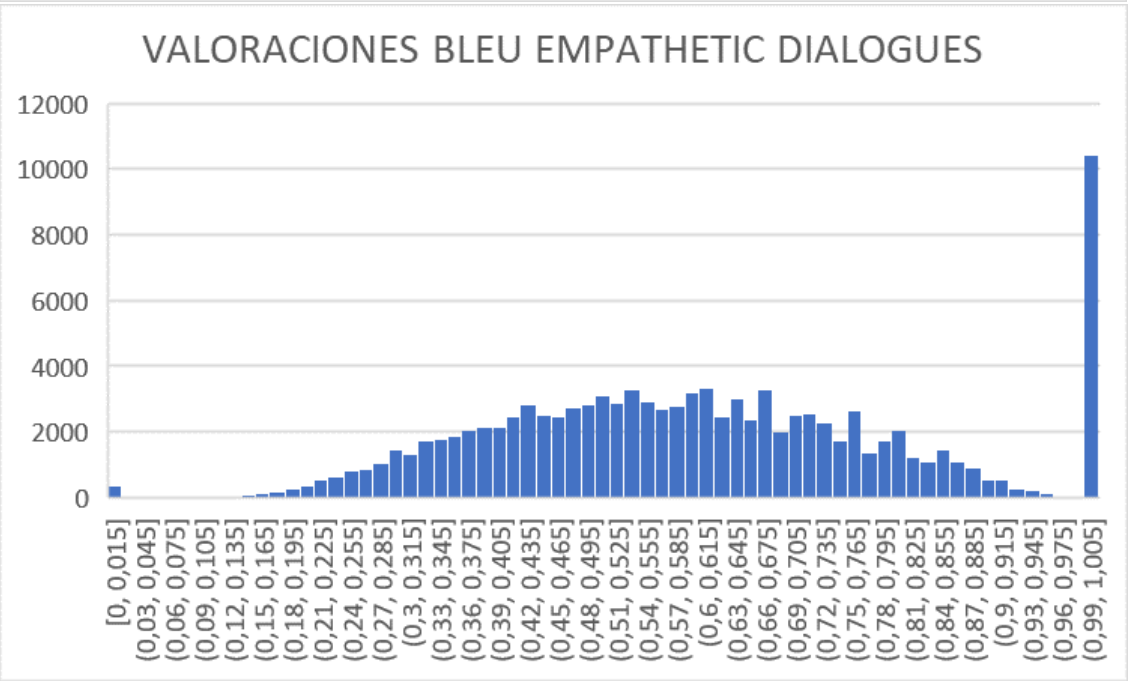




**ILUSTRACIÓN 61. Histogramas de las valoraciones de BLEU en DD y ED.**

En la gráfica se puede observar que las traducciones de ambas bases de datos muestran comportamientos similares al ser evaluadas con BLEU. A pesar de que el comportamiento de ambas bases de datos se asemeja, teniendo ambas el mayor número de valoraciones en la última franja, se observa que hay una diferencia considerable entre ambas. Para mostrar esta diferencia numéricamente se adjunta una tabla con la cantidad de valoraciones en ambas bases de datos en diferentes rangos y su porcentaje respecto al total de valoraciones.

| RANGO | DD (#) | DD (%) | ED (#) | ED (%) |
|---|---|---|---|---|
| **[0 - 0,5)** | 19.008 | 18,5 | 36.181 | 33,7 |
| **[0,5 – 0,7)** | 31.188 | 30,3 | 37.948 | 35,4 |
| **[0,7 – 0,8)** | 15.039 | 14,6 | 13.766 | 12,8 |
| **[0,8 – 0,9)** | 8.986 | 8,7 | 7.746 | 7,2 |
| **[0,9 – 1)** | 1.298 | 1,3 | 1.142 | 1,1 |
| **[1]** | 27.449 | 26,6 | 10.437 | 9,8 |

**TABLA 26. Representación numérica de las valoraciones de BLEU.**

Destaca la diferencia en la cantidad de valoraciones menores a 0,5, representando en ED un 15,2% más del total de sus valoraciones que en DD. También existe una diferencia considerable en la cantidad de puntuaciones perfectas, representando en DD un 26,6% del total mientras que tan solo un 9,8% en el total de valoraciones de ED. Se muestra también una tabla con la estadística descriptiva de las valoraciones enfrentadas.

| VARIABLE | VALOR DD | VALOR ED |
|---|---|---|
| **MEDIA** | 0.715 | 0.602 |
| **VARIANZA** | 0.049 | 0.043 |
| **DESVIACIÓN ESTÁNDAR** | 0.221 | 0.209 |
| **COEFICIENTE DE VARIACIÓN** | 0.308 | 0.347 |
| **VALOR MÍNIMO (XMÍN)** | 0 | 0 |
| **CUARTIL 1** | 0.541 | 0.447 |
| **MEDIANA (CUARTIL 2)** | 0.707 | 0.588 |
| **CUARTIL 3** | 1 | 0.74 |
| **VALOR MÁXIMO (XMÁX)** | 1 | 1 |
| **RANGO (XMÁX-XMÍN)** | 1 | 1 |

**TABLA 27. Estadística descriptiva de las valoraciones de BLEU para DD y ED.**

La estadística descriptiva muestra valores similares entre ambas bases de datos, aunque ligeramente superiores para DD, que tiene una media superior a ED. Los demás valores como la desviación estándar tienen valores muy similares en ambos casos. Se puede decir que la traducción de DD es ligeramente superior a ED respecto a la métrica BLEU.

La desviación estándar en ambos casos ronda el 0,2, lo cual indica cierto nivel de dispersión, dado que el rango de valores estudiados es de 0 a 1. La media y la mediana, sin embargo, son muy similares y describen la simetría de los resultados.

Se muestran a continuación ejemplos de oraciones con sus puntuaciones de BLEU que representan a los diferentes rangos de puntuación.

DAILYD-000599-0005. 1.0

No, it seems too old-fashioned.

No, parece demasiado anticuado.

No, it seems too old-fashioned.

DAILYD-001448-0006. 0.93

So what do you do, exactly?

Entonces, ¿qué haces exactamente?

So, what exactly are you doing?

DAILYD-003378-0010. 0.86

Okay, I guess I'll cook.

Bien, supongo que cocinaré.

Well, I guess I'll cook.

DAILYD-006762-0004. 0.70

You really don’t need to.

Realmente no es necesario.

It's really not necessary.

DAILYD-006524-0000. 0.66

Can you tell me a little about Paris?

¿Puedes contarme un poco sobre París?

Can you tell me a little bit about Paris?

DAILYD-001150-0004. 0.43

Which condiment do you use for?

¿Para qué condimento usas?

What seasoning do you use?

MPATHY-023924-0001. 1.0

What went wrong?

¿Qué salió mal?

What went wrong?

MPATHY-001719-0003. 0.92

yeah, you're hungry... it's there.

Sí, tienes hambre... está ahí.

Yes, you're hungry... it's there.

MPATHY-008230-0004. 0.83

I haven't been there since I was a child .

No he estado allí desde que era un niño.

I haven't been there since I was a kid .

MPATHY-016552-0002. 0.73

Hey! Good luck to you! That's got to be tough!

¡Oye! ¡Buena suerte! ¡Eso tiene que ser duro!

Hey! Good luck! That's got to be tough!

MPATHY-002474-0000. 0.66

Some kids aren't as fortunate as mine.

Algunos niños no son tan afortunados como los míos.

Some kids aren't as lucky as mine.

MPATHY-000543-0003. 0.34

I'm happy it went well for you!

¡Me alegro de que te haya ido bien!

I'm glad you did well!

Se han escogido oraciones breves para que sea más clara su lectura y comprensión. Se puede comprobar en estos ejemplos, como existen intervenciones con puntuaciones bajas que son traducciones correctas. Esto se debe a que la métrica BLEU no considera el contenido semántico en la evaluación de la traducción.

Se obtuvo también el número medio de caracteres por intervención, reflejado en la siguiente tabla:

| RANGO | ED | DD |
|---|---|---|
| **[0 – 0,5)** | 69,04 | 60,87 |
| **[0,5 – 0,7)** | 78,17 | 71,86 |
| **[0,7 – 0,8)** | 77,83 | 77,95 |
| **[0,8 – 0,9)** | 76,22 | 83,72 |
| **[0,9 – 1)** | 80,40 | 96,50 |
| **[1]** | 40,40 | 37,04 |

**TABLA 28. Media de caracteres por intervención por rangos en DD y ED.**

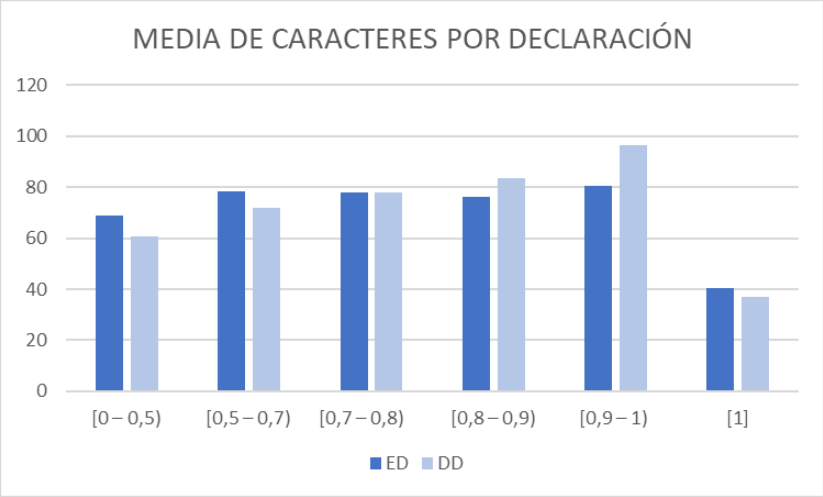



**ILUSTRACIÓN 62. Representación gráfica de la media de caracteres por intervención.**

Se observó que las intervenciones con puntuaciones perfectas eran notablemente menos extensas que el resto de los rangos valorados. Esto puede deberse a la dificultad que muestran los sistemas NMT a la hora de traducir oraciones extensas (de más de 20 palabras aproximadamente), especialmente si son más extensas que las empleadas en el entrenamiento del modelo de NMT [38].

Se realizó también el cálculo de la valoración BLEU de corpus. La valoración de corpus es especialmente interesante en bases de datos de tamaño considerable (de más de 2.000 oraciones) [39] porque la cantidad de datos recolectados es suficiente para realizar una estimación de confianza. Los resultados se muestran a continuación:

| | DD | ED |
|---|---|---|
| **CORPUS BLUE** | 0,695 | 0,592 |

**TABLA 29. Valoraciones BLEU de corpus.**

Ambas valoraciones son muy elevadas. Expresadas en porcentajes las puntuaciones serían 69,5 para DD y 59,2 para ED. Según la página de Google Cloud [32], estas

[32] https://cloud.google.com/translate/automl/docs/evaluate

evaluaciones son “Traducciones de calidad muy alta, adecuadas y fluidas” y “Calidad generalmente mejor que la humana” respectivamente.

### 4.1.2 Análisis de los resultados de Sentence Transformers

Ya que BLEU no evalúa las traducciones por su semántica, se buscaron modelos de última generación para evaluar las dos bases de datos. Los modelos de Sentence Transformers fueron los escogidos para evaluar ambas bases de datos.

Se analizó en primer lugar la base de datos DD.

Como se explica en el apartado 3.2.1, se optó por escoger un solo modelo para realizar un estudio más exhaustivo de los resultados. El criterio seguido para escogerlo fue que se asemejara al criterio humano. Para ello, se puntuaron manualmente 392 intervenciones con el siguiente resultado:

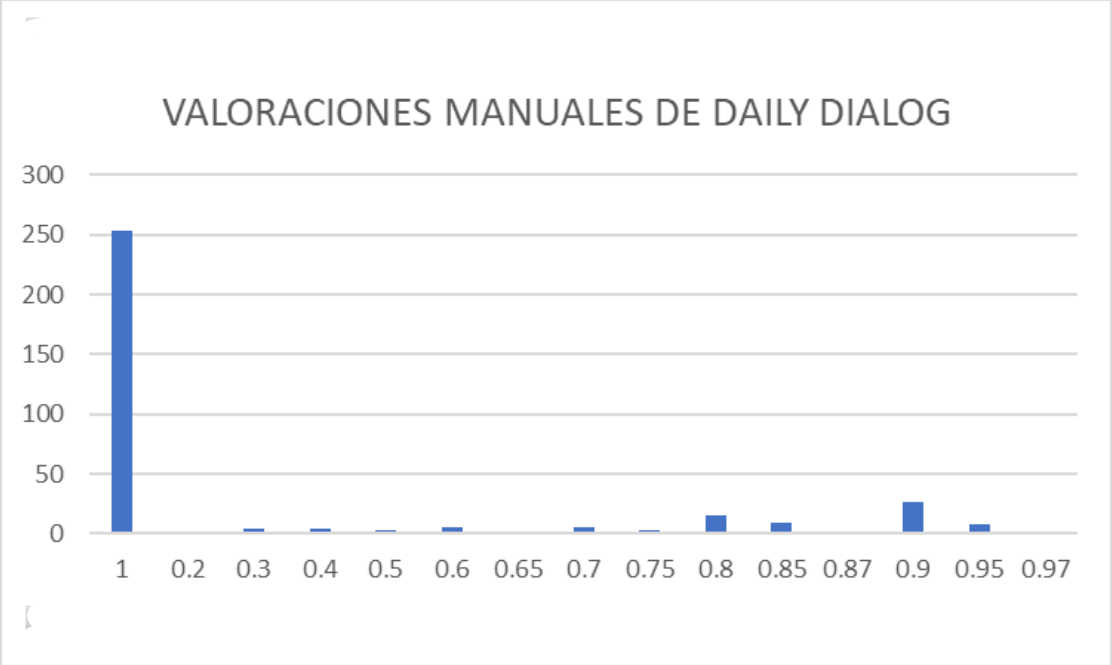



**ILUSTRACIÓN 63. Gráfico de valoraciones de las puntuaciones manuales.**

Como se puede comprobar, las valoraciones manuales presentan resultados muy satisfactorios. La media de las valoraciones fue igual a 0,82, un resultado elevado que indica la calidad de la base de datos estudiada.

| Valoración | Cuenta de Valoración |
|---|---|
| **1** | 253 |
| **0,2** | 2 |
| **0,3** | 4 |
| **0,4** | 4 |
| **0,5** | 3 |
| **0,6** | 6 |
| **0,65** | 1 |
| **0,7** | 6 |
| **0,75** | 3 |
| **0,8** | 16 |
| **0,85** | 9 |
| **0,87** | 1 |
| **0,9** | 26 |
| **0,95** | 8 |
| **0,97** | 1 |

**TABLA 30. Tabla con los resultados de las valoraciones manuales.**

Una vez obtenidos estos resultados, se obtuvieron las valoraciones de los seis modelos considerados, para escoger aquél que se asemejara más al vector de puntuaciones manuales.

Los resultados obtenidos fueron los siguientes:

| MODELO | SIMILITUD COSENO |
|---|---|
| **MODELO_0_MULTI** | 0,9813 |
| **MODELO_1_MULTI** | 0,9803 |
| **MODELO_2_MULTI** | 0,9824 |
| **MODELO_3_MULTI** | 0,9837 |
| **MODELO_4_MULTI** | 0,9803 |
| **MODELO_5_MULTI** | 0,9815 |

**TABLA 31. Valores de similitud coseno entre las valoraciones de los modelos de ST y las valoraciones manuales.**

Todos los modelos muestran gran similitud coseno con las valoraciones manuales. Esto nos indica que todos los modelos son apropiados para evaluar de manera pareja a un evaluador humano. Como el objetivo era encontrar un único modelo, se ordenaron los modelos de mejor a peor puntuación.

1. Modelo_3_multi
2. Modelo_2_multi
3. Modelo_5_multi
4. Modelo_0_multi
5. Modelo_1_multi y Modelo_4_multi

También se calculó el coeficiente de Pearson para tener otra medida de similitud. El resultado se muestra en la siguiente tabla.

| MODELO | COEF PEARSON |
|---|---|
| **MODELO_0_MULTI** | 0,26 |
| **MODELO_1_MULTI** | 0,24 |
| **MODELO_2_MULTI** | 0,27 |
| **MODELO_3_MULTI** | 0,30 |
| **MODELO_4_MULTI** | 0,22 |
| **MODELO_5_MULTI** | 0,14 |

**TABLA 32. Coeficientes Pearson entre los vectores obtenidos con los modelos de ST y las valoraciones manuales.**

En este caso, se encontró que solamente el Modelo_3_multi presentaba correlación moderada mientras que los otros modelos presentaban correlación débil.

Se optó por el **Modelo_3_multi: *paraphrase-xlm-r-multilingual-v1.*** Se trata de un modelo que presenta un rendimiento superior a otros modelos en la tarea de la similitud semántica textual [36].

Este modelo, pasó a considerarse el modelo base para el resto de las evaluaciones.

Se van a mostrar los resultados obtenidos al evaluar ambas bases de datos con el Modelo_3_multi.

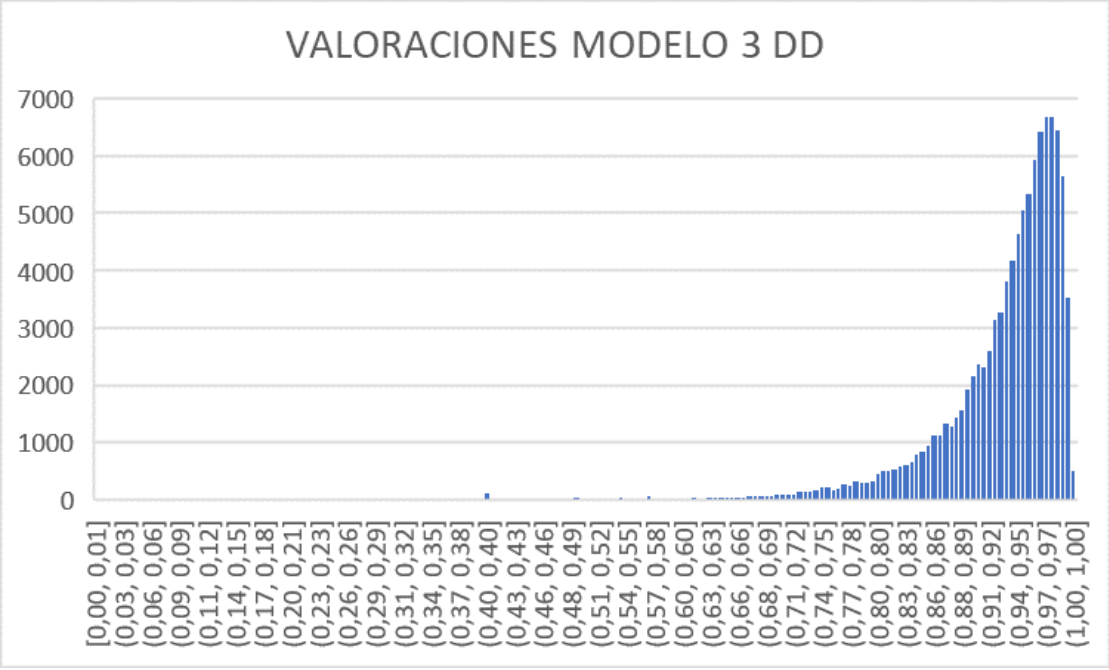



**ILUSTRACIÓN 64. Histograma de los resultados obtenidos con el Modelo_3_multi para DD.**

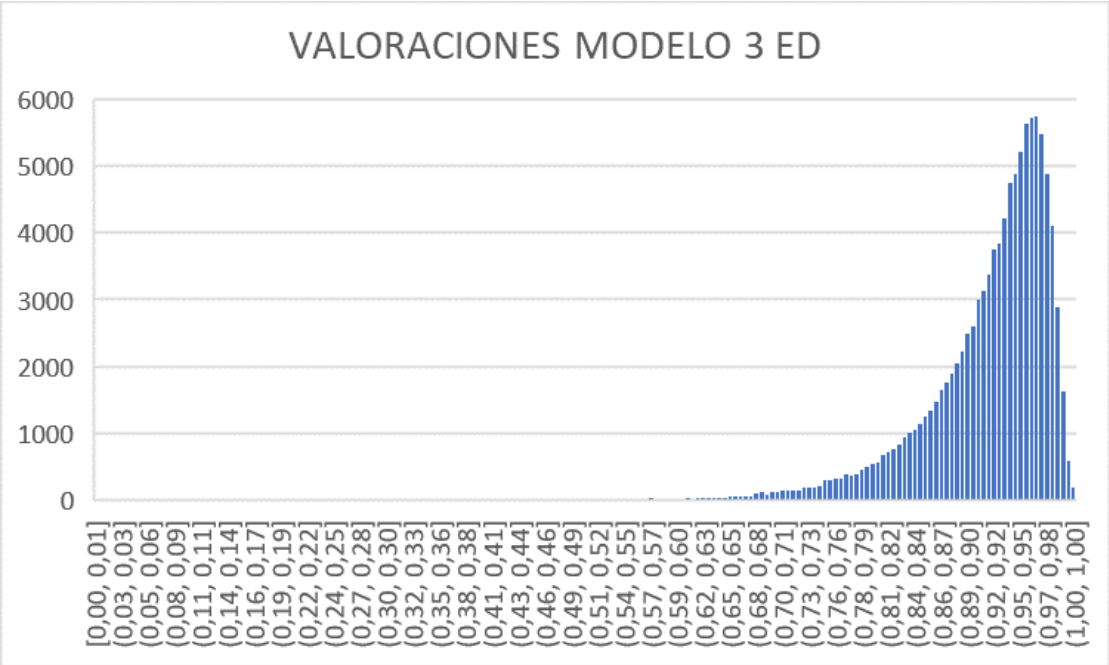



**ILUSTRACIÓN 65. Histograma de los resultados obtenidos con el Modelo_3_multi para ED.**

Los histogramas muestran la similitud entre las evaluaciones de los dos modelos. Para realizar un análisis de la estadística descriptiva y comparar las bases de datos con mayor facilidad, se ha realizado una tabla con los resultados de ambas.

| VARIABLE | VALOR DD | VALOR ED |
|---|---|---|
| **MEDIA** | 0,923 | 0,912 |
| **VARIANZA** | 0,006 | 0,005 |
| **DESVIACIÓN ESTÁNDAR** | 0,077 | 0,074 |
| **COEFICIENTE DE VARIACIÓN** | 0.083 | 0,081 |
| **VALOR MÍNIMO (XMÍN)** | 0 | 0 |
| **CUARTIL 1** | 0,904 | 0,886 |
| **MEDIANA (CUARTIL 2)** | 0,947 | 0,932 |
| **CUARTIL 3** | 0,972 | 0,960 |
| **VALOR MÁXIMO (XMÁX)** | 1 | 1 |
| **RANGO (XMÁX-XMÍN)** | 1 | 1 |

**TABLA 33. Estadística descriptiva de las valoraciones del Modelo_3_multi para DD y ED.**

Los resultados en este caso son muy similares para ambas bases de datos, a diferencia de los resultados obtenidos con la métrica BLEU. La media se sitúa en 0,923 para DD y en 0,912 para ED. Son resultados muy elevados, por lo que podemos considerar que las oraciones originales y las traducciones tienen significados semánticos muy similares.

La media y la mediana se encuentran muy próximas tanto para DD como para ED, lo que nos indica la simetría de los datos. Además, la desviación estándar ronda el 0,07 en ambas bases de datos, lo que indica que las valoraciones se concentran alrededor de la media. Teniendo en cuenta lo elevado que es el valor de la media, se pueden considerar unos resultados muy buenos.

Se dividió la base de datos en grupo de análisis según las valoraciones obtenidas para facilitar su estudio. A continuación, se nombrarán los grupos de estudio con las valoraciones que representan.

- Grupo A. Puntuaciones $p \leq 0.6$.
- Grupo B. Puntuaciones $0.6 < p < 0.7$.
- Grupo C. Puntuaciones $0.7 < p \leq 0.8$.
- Grupo D. Puntuaciones $0.8 < p \leq 0.9$.
- Grupo E. Puntuaciones $0.9 < p \leq 0.95$.
- Grupo F. Puntuaciones $0.95 < p \leq 1$.

Se representan el número de intervenciones en cada grupo y su frecuencia relativa respecto al total de cada base de datos.

| GRUPO | FRECUENCIA ABSOLUTA DD | FRECUENCIA ABSOLUTA ED | FRECUENCIA RELATIVA DD | FRECUENCIA RELATIVA ED |
|---|---|---|---|---|
| **A** | 889 | 595 | 0,863% | 0,554% |
| **B** | 1.071 | 1.279 | 1,040% | 1,193% |
| **C** | 3.864 | 5.621 | 3,752% | 5,242 % |
| **D** | 18.296 | 25.709 | 17,769% | 23,977% |
| **E** | 29.948 | 36.690 | 29,085% | 34,222% |
| **F** | 48.898 | 37.324 | 47,491% | 34,812% |

**TABLA 34. Tabla de frecuencias absoluta y relativa de cada grupo de estudio para DD y ED.**

Se observa en la tabla que DD tiene un porcentaje mayor de puntuaciones en el grupo F, el de las puntuaciones más elevadas. En cualquier caso, hay que tener en cuenta que el 94,345% y el 93,011% de las valoraciones de las traducciones de DD y ED respectivamente se encuentran por encima de 0,8.

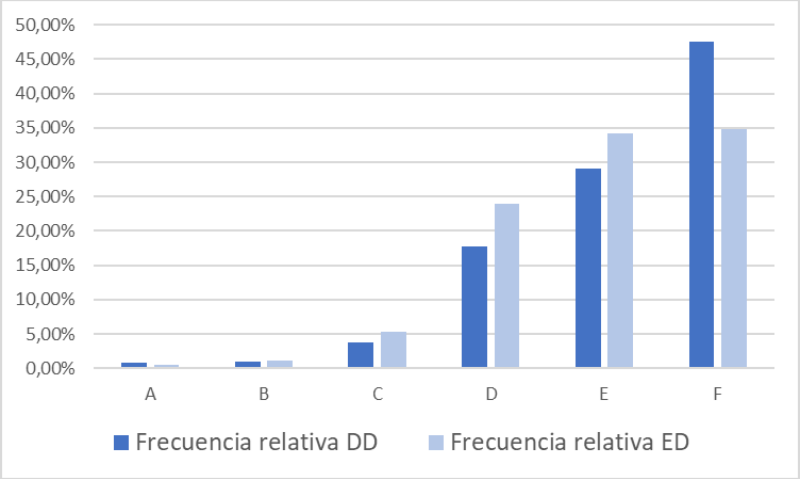



**ILUSTRACIÓN 66. Gráfico de columnas de las frecuencias relativas de DD y ED.**

También se ha contado el número de caracteres en cada grupo con el siguiente resultado.

| RANGO | ED | DD |
|---|---|---|
| **GRUPO A** | 34,73 | 54,14 |
| **GRUPO B** | 42,21 | 44,80 |
| **GRUPO C** | 47,86 | 51,08 |
| **GRUPO D** | 58,28 | 64,97 |
| **GRUPO E** | 69,48 | 74,92 |
| **GRUPO F** | 60,79 | 66,74 |

**ILUSTRACIÓN 67. Media de caracteres por intervención por rangos en DD y ED.**

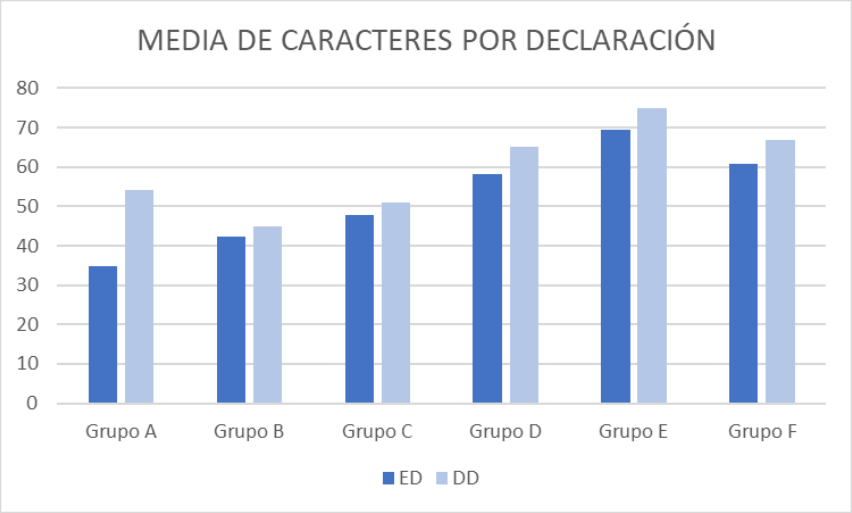



**ILUSTRACIÓN 68. Representación gráfica de la media de caracteres por intervención.**

Ambas bases de datos muestran un comportamiento muy similar, siendo el grupo E el que contiene las intervenciones más extensas.

Se comprobó también la media de caracteres en las valoraciones que recibieron una puntuación perfecta. La media de DD era de 11,85 caracteres y la de ED de 4,46. Al tratarse de un modelo multilingüe, es más complicado obtener dos embeddings que sean perfectamente equivalentes.

Las oraciones que reciben una puntuación perfecta en ambos casos son intervenciones con nombres propios, valores numéricos e interjecciones.

Se revisaron manualmente oraciones aleatorias de cada grupo de estudio. Los errores encontrados se detallan en los apartados 3.2.1 y 3.2.2 de este documento.

Una vez procesados los datos y se redondearon, se obtuvieron los siguientes resultados.

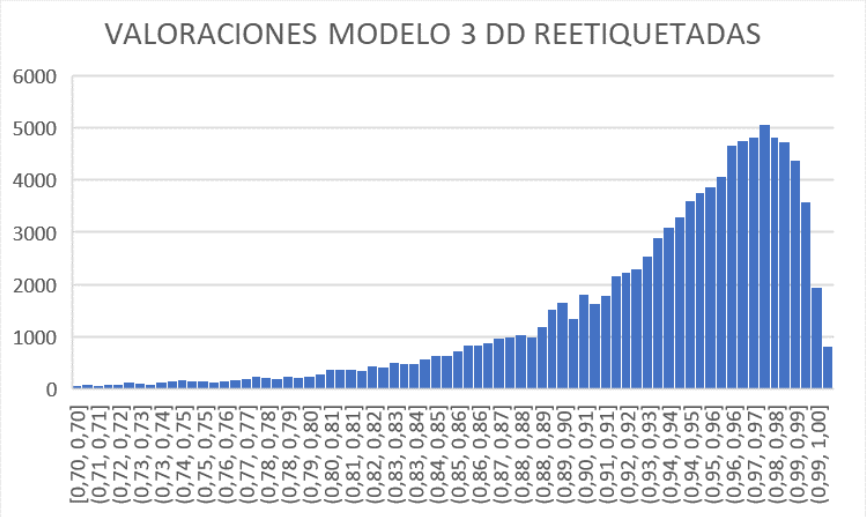



**ILUSTRACIÓN 69. Histograma de las valoraciones del Modelo_3_multi tras el reetiquetado de 0,7 a 1 de DD.**

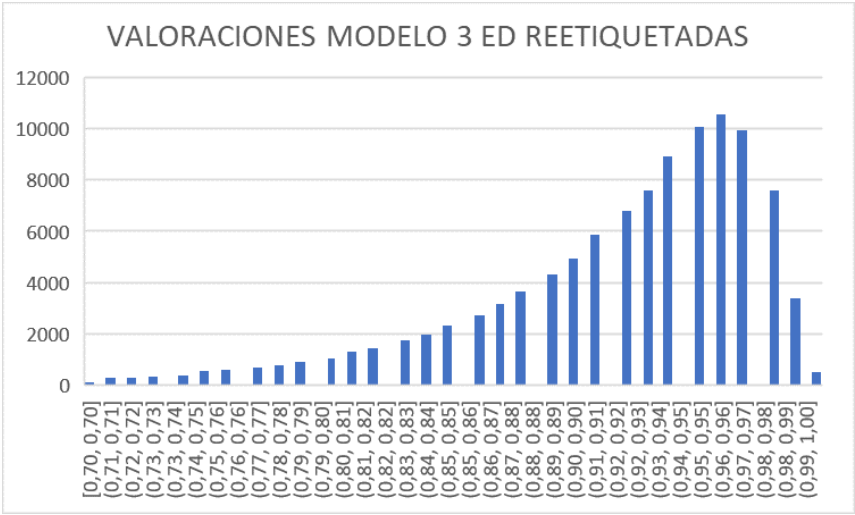



**ILUSTRACIÓN 70. Histograma de las valoraciones del Modelo_3_multi tras el reetiquetado de 0,7 a 1 de ED.**

| VARIABLE | VALOR DD | VALOR ED |
|---|---|---|
| **MEDIA** | 0.932 | 0.917 |
| **VARIANZA** | 0.003 | 0.003 |
| **DESVIACIÓN ESTÁNDAR** | 0.056 | 0.058 |
| **COEFICIENTE DE VARIACIÓN** | 0.060 | 0.063 |
| **VALOR MÍNIMO (XMÍN)** | 0.7 | 0,7 |
| **CUARTIL 1** | 0.908 | 0.89 |
| **MEDIANA (CUARTIL 2)** | 0.948 | 0.93 |
| **CUARTIL 3** | 0.973 | 0.96 |
| **VALOR MÁXIMO (XMÁX)** | 1 | 1 |
| **RANGO (XMÁX-XMÍN)** | 0.3 | 0,3 |

**TABLA 35. Descripción estadística de las valoraciones de Modelo_3_multi tras el reetiquetado entre 0.7 y 1 de DD y ED.**

Tras el procesado, la media aumenta ligeramente y el rango pasa de ser 1 a 0,3. La desviación estándar disminuye en 0,02 aproximadamente para ambas bases de datos. por lo tanto, las valoraciones se concentran más cerca de la media tras el procesado.

Como conclusión, se puede determinar que las bases de datos son de calidad, ya que reciben tanto en BLEU como en el Modelo_3_multi una valoración muy elevada.

### 4.1.3 Análisis de los resultados de la regresión lineal

Para conectar la métrica de base en el ámbito de la investigación, BLEU y la evaluación del modelo XLM-R de Sentence Transformers (Modelo_3_multi) se entrenó un modelo de regresión lineal.

Primero, se evaluó un modelo de dos entradas, modelos multilingües, escogidos según su similitud con la evaluación humana, que se calculó para la evaluación con modelo simple de ST. Se buscaba que la salida estuviera relacionada con BLEU, por tanto, se buscó aquel modelo monolingüe de ST que se asemejara más a las valoraciones obtenidas por esta métrica.

La obtención del modelo de salida se realizó de la misma manera que en el análisis con un modelo simple, mediante la comparación de la similitud de los vectores de resultados de los diferentes modelos.

Se escogieron los cinco modelos monolingües con mejor rendimiento de la web de Sentence Transformers y se obtuvieron los vectores de valoraciones para la base de datos DD. Los resultados se compararon con el vector de resultados de BLEU.

Los resultados de la similitud coseno fueron los siguientes.

| MODELO | SIMILITUD COSENO |
|---|---|
| **MODELO_0_MONO** | 0.9653 |
| **MODELO_1_MONO** | 0.9668 |
| **MODELO_2_MONO** | 0.9663 |
| **MODELO_3_MONO** | 0.9665 |
| **MODELO_4_MONO** | 0.9666 |

**TABLA 36. Valores de similitud coseno entre los modelos de ST y las valoraciones de BLEU.**

Todos los resultados mostraron gran similitud con las valoraciones de BLEU. Se ordenan los resultados de mayor a menor puntuación a pesar de que son muy similares, obteniendo el siguiente resultado.

1. Modelo_1_mono
2. Modelo_4_mono
3. Modelo_3_mono
4. Modelo_2_mono
5. Modelo_0_mono

Se obtuvo también el coeficiente de Pearson para comparar la correlación de los vectores de puntuaciones de los modelos de ST y de las evaluaciones de BLEU. El resultado fue el siguiente.

| MODELO | COEF PEARSON |
|---|---|
| MODELO_0_MONO | 0,497 |
| MODELO_1_MONO | 0,528 |
| MODELO_2_MONO | 0,518 |
| MODELO_3_MONO | 0,531 |
| MODELO_4_MONO | 0,526 |

**TABLA 37. Coeficientes Pearson entre los vectores obtenidos con los modelos de ST y las valoraciones manuales.**

En este caso, se encontró una correlación fuerte entre las valoraciones de los modelos y BLEU. Se ordenaron también los resultados quedando.

1. Modelo_3_mono
2. Modelo_1_mono
3. Modelo_4_mono
4. Modelo_2_mono
5. Modelo_0_mono

Finalmente se escogió el **Modelo_1_mono: *paraphrase-TinyBERT-L6-v2***, ya que obtuvo la mejor puntuación en la similitud coseno y la segunda mejor puntuación del coeficiente de correlación de Pearson.

Una vez escogida la salida de entrenamiento del modelo, se escogieron los dos modelos de entrada. Se escogieron el **Modelo_2_multi: *distiluse-base-multilingual-cased-v1*** y **Modelo_3_multi: *paraphrase-xlm-r-multilingual-v1*** porque obtuvieron las mejores puntuaciones respecto a la evaluación humana.

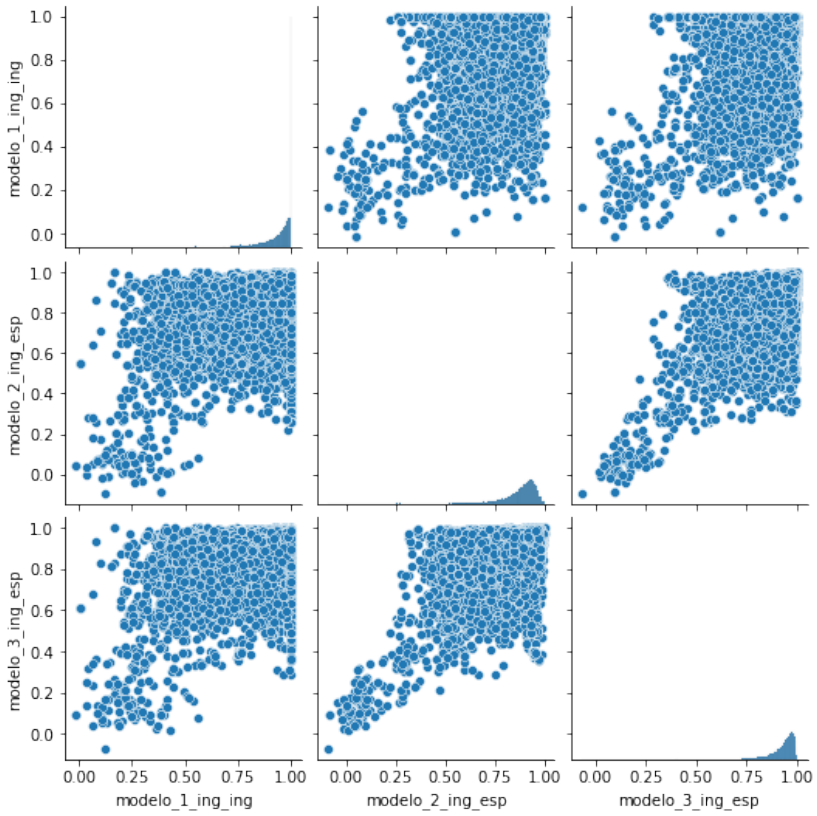



**ILUSTRACIÓN 71. Diagramas de dispersión entre los modelos Modelo_1_mono, Modelo 2_multi y Modelo_3_multi.**

Los diagramas de dispersión muestran la relación entre variables. En este caso, todas las variables muestran una relación positiva y lineal. Es decir, los valores crecientes de un modelo están asociados con los valores crecientes de otro. Los modelos multilingües muestran una relación más fuerte entre ellos que la que muestran con el Modelo_1_mono.

Los resultados obtenidos tras el entrenamiento del modelo fue el siguiente vector de coeficientes: `[0.16543127 0.5831466]`.

Se representa en el siguiente diagrama de dispersión la relación entre las predicciones del modelo entrenado y los datos reales del Modelo_1_mono separados para el test.

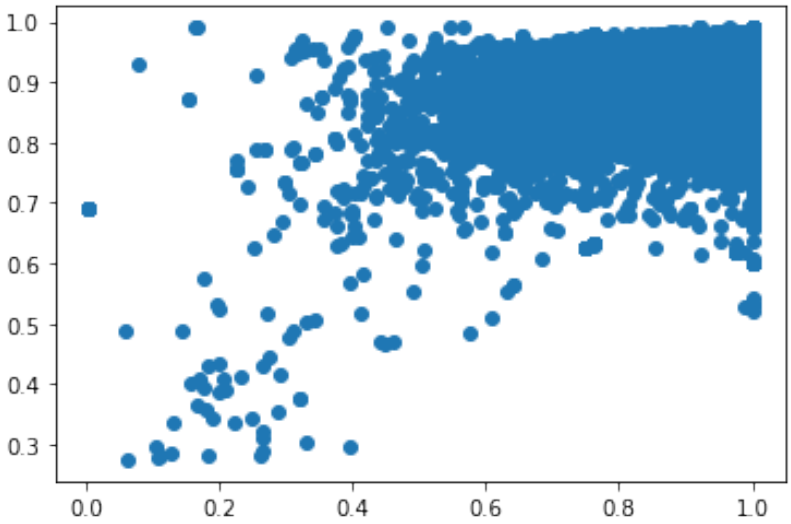



**ILUSTRACIÓN 72. Diagrama de dispersión del modelo de LR entrenado con 2 entradas.**

El resultado no presenta una gran linealidad y debido a la acumulación de datos en las puntuaciones más altas es complicado valorar el modelo sin más información.

Se utilizó la función incorporada en Sklearn para evaluar el coeficiente de determinación. Se obtuvo un 0,266, lo cual significa que se explica un 26,6% de los resultados.

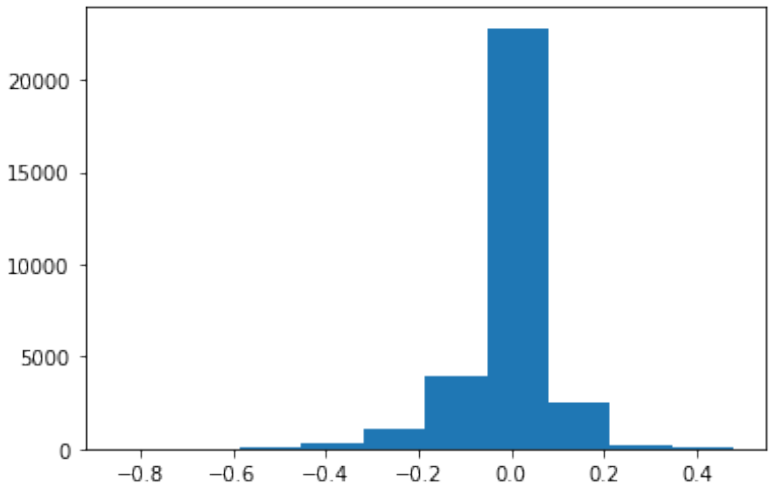



**ILUSTRACIÓN 73. Gráfica de residuos de las predicciones del modelo LR entrenado con 2 entradas, y las salidas de test.**

La gráfica de residuos sin embargo proporcionó una respuesta más prometedora, ya que los residuos se concentran alrededor del 0, especialmente entre los valores -0,2 y 0,2.

Teniendo en cuenta los valores reducidos de la desviación estándar para los modelos estudiados se procedió a realizar un test manual.

Se calcularon los porcentajes de acierto según distintas desviaciones relativas entre la predicción y el valor real de los datos de test separados.

| Desviación típica relativa | Porcentaje de acierto |
|---|---|
| **0,02** | 22,61% |
| **0,03** | 37,76% |
| **0,04** | 51,16% |
| **0,05** | 61,29% |
| **0,06** | 68,45% |
| **0,07** | 74% |
| **0,08** | 78,28% |

**TABLA 38. Representación de porcentajes de acierto según la desviación típica relativa entre test y predicción de LR de 2 entradas.**

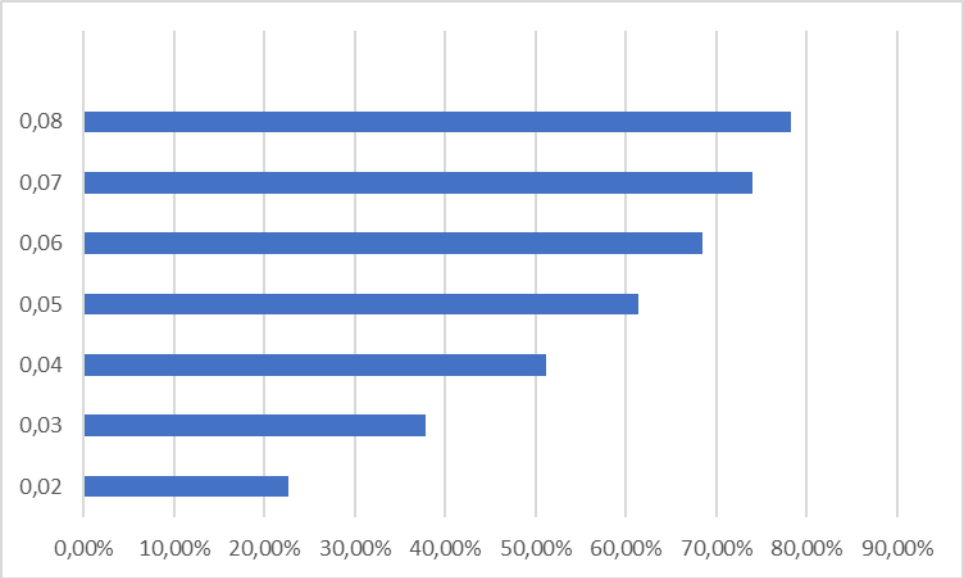



**ILUSTRACIÓN 74. Porcentajes de acierto según la desviación típica relativa entre test y predicción de la LR con 2 entradas.**

Con una desviación típica relativa de 0,07, el porcentaje de acierto es del 74%. Este es un resultado muy satisfactorio ya que esta arquitectura es novedosa y no se ha empleado nunca para la evaluación de modelos de MT.

Se realizó también un modelo de LR con los 6 modelos multilingües como entrada.

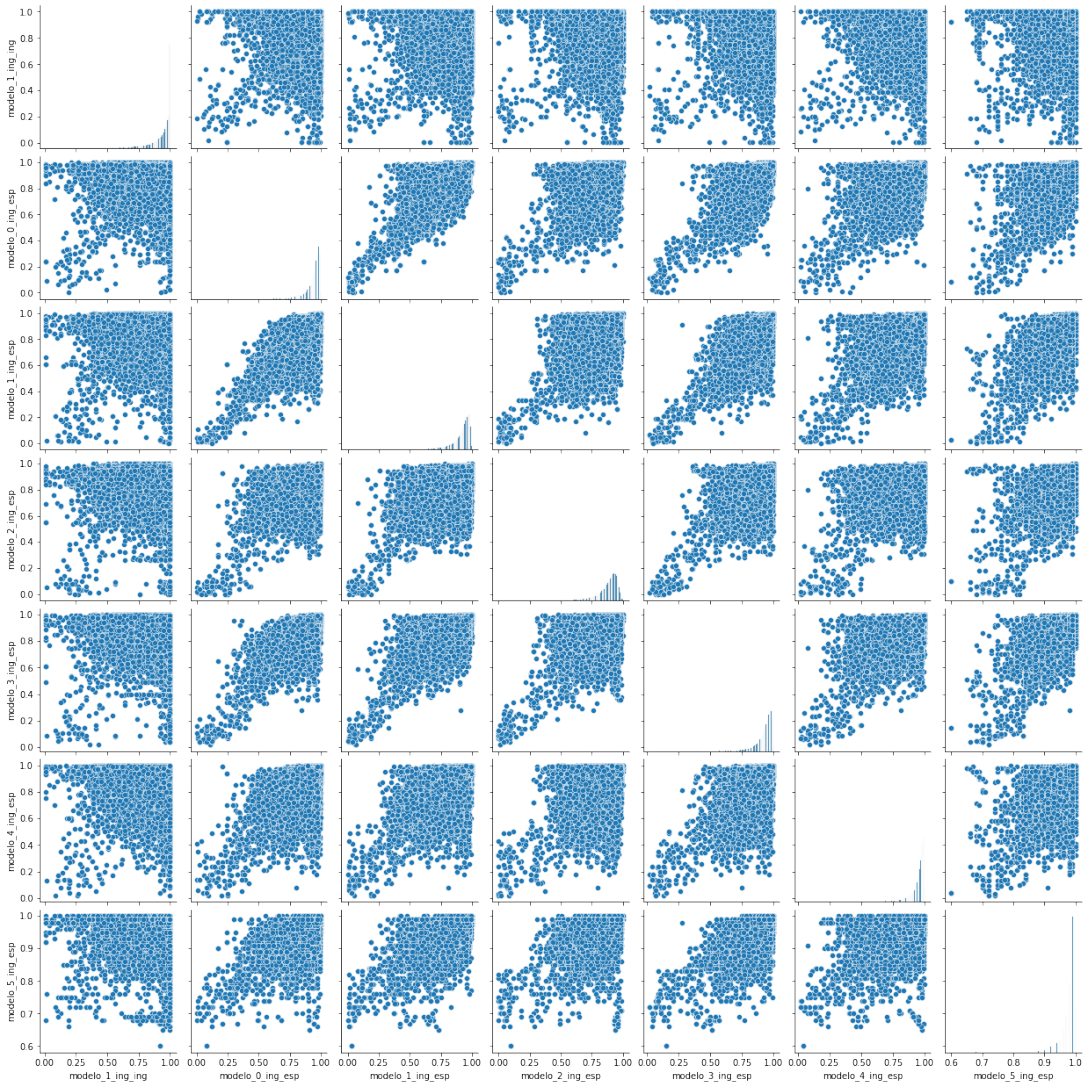



**ILUSTRACIÓN 75. Diagramas de dispersión entre los diferentes modelos.**

Como se puede ver en la figura, todos los diagramas presentan una correlación positiva y lineal, más o menos fuerte según la pareja considerada.

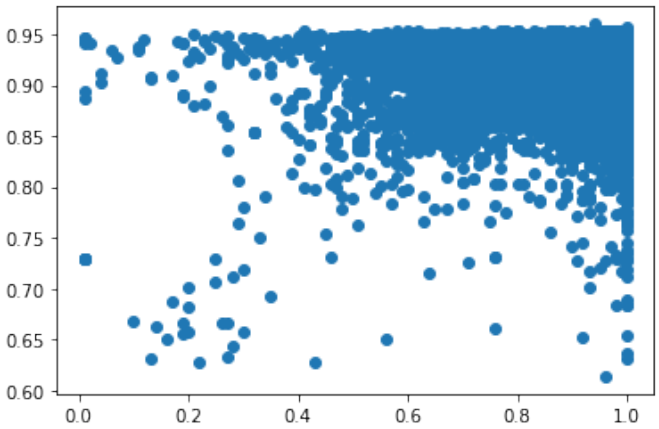



**ILUSTRACIÓN 76. Diagrama de dispersión del modelo de LR entrenado con 6 entradas.**

El modelo entrenado obtuvo el siguiente vector de coeficientes: `[0.21823429 0.01476081 0.01544197 0.13443293 -0.00339057 -0.09809735]`.

Como se puede ver en la figura del diagrama de dispersión, los datos se concentran en los valores más elevados, por lo que es complicado determinar la correlación entre los mismos. Se procedió de la misma manera que con el modelo de LR entrenada con los entradas, realizando un test manual. La gráfica de residuos muestra también los valores centrados en el 0.

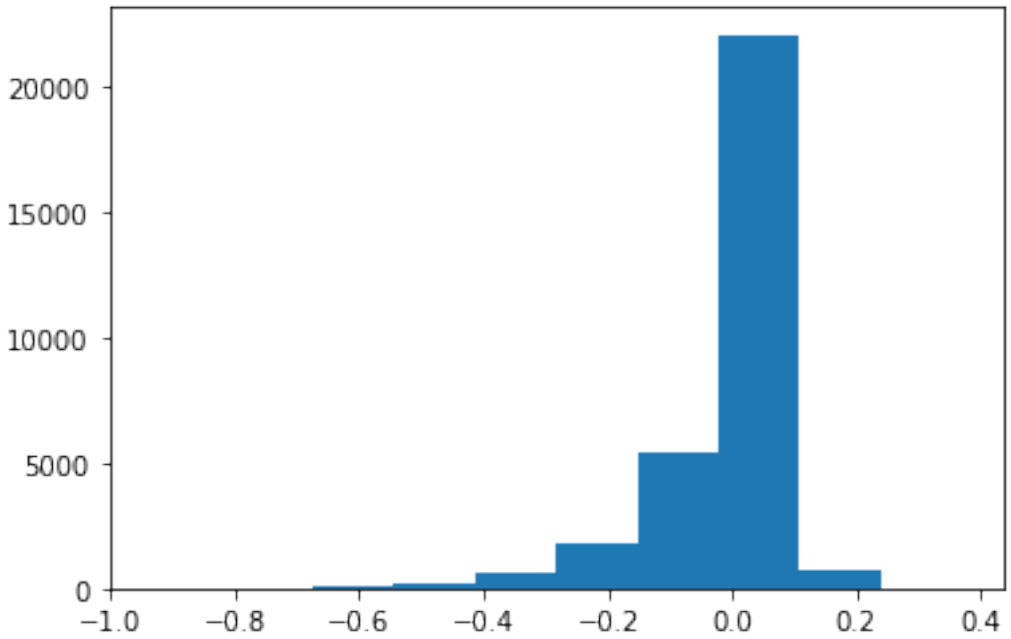



**ILUSTRACIÓN 77. Gráfica de residuos de las predicciones del modelo LR entrenado con 6 entradas, y las salidas de test.**

| Desviación típica relativa | Porcentaje de acierto |
|---|---|
| **0,02** | 14,25% |
| **0,03** | 22,24% |
| **0,04** | 31,14% |
| **0,05** | 41,96% |
| **0,06** | 59,23% |
| **0,07** | 70,08% |
| **0,08** | 77% |

**TABLA 39. Representación de los diferentes porcentajes de acierto según la desviación típica relativa entre predicción del modelo de 6 entradas y el valor real.**

Para el modelo de 6 entradas se encuentra un resultado similar al modelo de 2 entradas, aunque un poco menos efectivo. En este caso, con la misma desviación típica relativa que en el modelo de 6 entradas, se obtiene un porcentaje de acierto del 70,08. Es un resultado algo inferior, pero sigue superando el 70%, lo cual nos indica que se trata de un modelo efectivo a la hora de predecir un resultado.

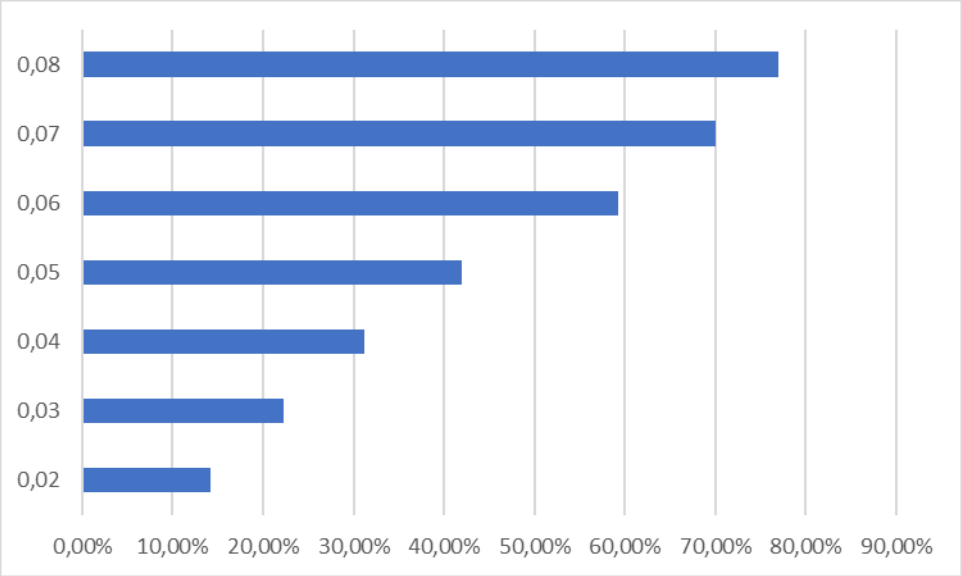



**ILUSTRACIÓN 78. Porcentajes de acierto según la distancia entre test y predicción de la LR con 6 entradas.**

La representación gráfica comparada con el modelo de 2 entradas muestra que para distancias menores a 0,7 es notablemente menos efectiva que el modelo de 2 entradas.

## 4.2 Discusión

Una vez analizados los resultados obtenidos, se pueden asegurar por un lado la calidad de ambas bases de datos tanto sintáctica como semántica y el resultado satisfactorio del entrenamiento de una arquitectura novedosa.

Esta arquitectura, es interesante ya que sirve para la obtención de valoraciones basadas en la métrica de base BLEU, pero con la adición de la evaluación semántica y de contexto. La novedad de esta arquitectura reside en que no precisa de traducciones candidatas en inglés por lo que se pueden comparar directamente las oraciones originales en inglés con sus traducciones en español para obtener una puntuación. Se podría emplear para otras parejas de lenguas, pero no se podría asegurar un resultado satisfactorio ya que no se han realizado pruebas con otras lenguas que no fueran en inglés-español.

Respecto a las traducciones de las bases de datos, se pueden comentar varias cosas.

Por un lado, las valoraciones de BLEU originalmente se pensaron para evaluar las oraciones traducidas con varias oraciones candidatas traducidas por expertos en la materia. Teniendo en cuenta que BLEU tiene más de 15 años, se ha ido variando su uso. Hoy en día, se emplea como una base para unificar y poder comparar los resultados de los ensayos que se realizan dentro del campo de la investigación del NLP. Sin embargo, se emplea generalmente una sola candidata para realizar la evaluación BLEU. Además, en este caso, se ha empleado como candidata una segunda traducción automática. El hecho de realizar una segunda traducción puede dar lugar a diversos escenarios: Una traducción correcta que pierda su significado con la segunda traducción, dando lugar a un falso negativo; Una traducción incorrecta, por ejemplo por una traducción literal, que se traduzca de vuelta a la lengua original de manera que se

asemeje a la original, obteniéndose un falso positivo; Una traducción correcta que se traduce de manera correcta de vuelta a la lengua original, positivo verdadero; Una traducción incorrecta que se traduce de vuelta perdiendo el sentido original, falso verdadero. Tras el análisis de las bases de datos, cabe destacar que se han encontrado de manera manual supervisada, es decir, analizando muestras aleatorias de conjuntos de datos, falsos positivos en BLEU, principalmente por traducciones literales de expresiones idiomáticas. Sin embargo, la evaluación positiva de la métrica concuerda con las traducciones de calidad que se han obtenido en ambos modelos.

Las valoraciones del modelo XLM-R, son especialmente interesantes, ya que no precisa de oraciones candidatas, sino que directamente se compara la oración original con su traducción. Los resultados obtenidos por este modelo resultan por lo general fiables según las revisiones que se han realizado de oraciones aleatorias en distintos rangos de puntuaciones. El caso que es más difícil de detectar por el modelo es el de las expresiones idiomáticas y frases hechas. Se han encontrado, especialmente en DD, expresiones poco frecuentes que el traductor de Azure no ha sido capaz de traducir correctamente, realizando una traducción literal. En muchos casos, las oraciones de este tipo obtienen puntuaciones más bajas que la media y suelen rondar el 0,7. Sin embargo, hay casos en los que las traducciones literales se dan por correctas y el modelo encuentra problemas a la hora de evaluar la semántica.

Algunos de los errores más recurrentes se han detectado y se pueden modificar fácilmente con un programa sencillo de procesado de los datos.

Como se ha dicho anteriormente, a pesar de que existen algunos casos de falsos positivos, ambas bases de datos cumplen con las condiciones para entrenar a un agente de conversación en español.

# Capítulo 5. Conclusiones y líneas futuras

Finalizado el proyecto, se ha tratado de reflejar en este capítulo las conclusiones a las que se ha llegado atendiendo a las diferentes etapas que ha vivido el trabajo. Además, se incluyen líneas de desarrollo futuro que se podrían estudiar a raíz del proyecto.

## 5.1 Conclusiones

Durante el desarrollo del proyecto, se han leído artículos, foros y capítulos de libros dedicados al NLP. Se trata de una rama de la Inteligencia Artificial que inunda todos los ámbitos de la vida y que está en un acelerado y constante crecimiento y evolución. Se han leído artículos de investigación del estado del arte y se han encontrado cambios, nuevas herramientas y modelos que han ido originándose al mismo tiempo que se desarrollaba el proyecto.

De hecho, los modelos multilingües empleados son muy recientes, no llegan ni siquiera al año y, sin embargo, desde el momento en que se comenzaron a emplear para realizar ensayos hasta el día en que se escribe este capítulo han surgido nuevos modelos que superan en mayor o menor medida el rendimiento de los predecesores [40, 41]. Google, Amazon, Facebook y otras gigantes invierten una gran cantidad de recursos económicos en el ámbito del NLP y ponen las herramientas a disposición de cualquier usuario. De esta manera se está creando una enorme red de constante mejora en las tecnologías existentes.

En cuanto a las herramientas de Traducción Automática, cabe destacar la calidad que han alcanzado en los últimos años con la mejora de los modelos neuronales, obteniéndose traducciones de alto nivel, comparables en algunos casos a las de traductores profesionales. Sin embargo, aún les queda un largo recorrido, sobre todo a la hora de considerar el contexto para la tarea de desambiguación.

Los resultados obtenidos en el proyecto han sido satisfactorios, consiguiendo desarrollar una arquitectura novedosa. Esta arquitectura, a pesar de tener margen de mejora, obtiene buenos resultados, especialmente en el modelo de 2 entradas.

Además, se han podido evaluar las traducciones de dos de las bases de datos más potentes a nivel emocional que se pueden encontrar para el entrenamiento de agentes conversacionales. Las traducciones realizadas con la herramienta de traducción de Azure, son de calidad y se ha podido medir gracias a los modelos de ST que se han empleado en el proyecto. Para su uso y explotación, se recomienda un procesado para deshacerse de los errores recurrentes encontrados y expuestos en el documento.

Otra de las conclusiones que se han obtenido al realizar este proyecto, es la necesidad por parte de toda la comunidad que investiga el NLP de avanzar más allá de la métrica BLEU. Se encuentran múltiples artículos [42, 43, 44] que exponen que BLEU ha quedado obsoleta y ha sido superada con creces por otras métricas de evaluación. Sin embargo, es evidente que se trata de una métrica sencilla y conveniente que será difícil de sustituir por su asentamiento dentro del marco del procesamiento del lenguaje natural.

Como conclusión acerca del trabajo realizado, cabe destacar que se ha podido aprender el lenguaje de programación Python, empleado en multitud de ámbitos y con una gran versatilidad, consiguiendo implementar programas de diversa índole. Se han cumplido los objetivos planteados para el proyecto y se ha obtenido una arquitectura novedosa con resultados satisfactorios.

## 5.2 Líneas futuras

En el artículo 'Six Challenges for Neural Machine Translation' [45] se hace referencia a algunos de los retos que se presentan en la MT. Entre ellos se encuentran las traducciones fuera de dominio y las palabras de uso poco frecuente.

### 5.2.1 Base de datos de expresiones idiomáticas y verbos frasales

Como se comentó en el capítulo 3 del documento, se encontraron errores recurrentes con las expresiones idiomáticas y frases hechas. Al tratarse el inglés de un idioma muy rico en verbos frasales y expresiones idiomáticas que, además se emplean de manera habitual, una línea de futuro podría ser la creación de una base de datos que contuviera un gran conjunto de este tipo de combinaciones que se pudieran emplear o bien para el procesamiento de las traducciones o bien para su evaluación.

### 5.2.2 Sistema de posprocesamiento

Otra línea de futuro para este proyecto sería el de crear un sistema de procesamiento, que tuviera en cuenta todos los errores que se presentan con frecuencia en las intervenciones traducidas, para obtener traducciones procesadas de una calidad aún más elevada. Esto sería relativamente sencillo de obtener mediante las herramientas que proporciona el lenguaje de programación Python.

### 5.2.3 Mejora de la arquitectura LR entrenada

Aunque se ha obtenido un resultado satisfactorio en el proyecto, al tratarse de una arquitectura novedosa no obtiene resultados de alto nivel. Por ello, se trabajaría en mejorar la arquitectura con más datos de entrenamiento, especialmente con oraciones incorrectas con puntuaciones bajas para que el modelo aprendiera a diferenciar estos casos con más soltura. También se podría probar la arquitectura con otras parejas de idiomas que no fueran inglés-español para evaluar su rendimiento

### 5.2.4 Contextualización

Una mayor contextualización en los embeddings mejora notablemente el rendimiento de las diferentes tareas del NLP [46]. En las oraciones traducidas se ha encontrado en algunos casos una descontextualización de intervenciones de una misma conversación

debido a que se han traducido de manera independiente. Como línea de futuro sería interesante que las intervenciones de una misma conversación se tradujeran juntas para tratar de obtener una mayor contextualización y con ello, un rendimiento mayor en las traducciones.

### 5.2.5 Uso de modelos en español

Otra línea que se podría tomar sería la de emplear un modelo monolingüe en español como el que se presenta en el artículo “Spanish Pre-Trained Bert Model And Evaluation Data” [47]. A la hora de emplear los embeddings habría que tener en cuenta la necesidad de mapear los modelos monolingües para que estuvieran en el mismo espacio vectorial y poder emplearlos para evaluación.

# ORGANIZACIÓN DEL PROYECTO

# 1 Gestión del proyecto

## 1.1 Ciclo de vida

Este trabajo se ha organizado en seis bloques, cada uno con los pertinentes paquetes de trabajo. Se numeran a continuación:

1. Definición y gestión del proyecto.
   1.1. Planificación del proyecto.
   1.2. Seguimiento del proyecto.
   1.3. Redacción del documento escrito.
2. Estudios previos.
   2.1. Estudio sobre el procesamiento del lenguaje natural.
   2.2. Estudio sobre el machine learning.
   2.3. Estudio sobre la traducción automática.
   2.4. Estudio sobre las métricas de evaluación automáticas.
   2.5. Estudio sobre los modelos de Sentence Transformers.
3. Diseño de programas.
   3.1. Selección de las herramientas de software.
   3.2. Familiarización con el lenguaje de programación Python.
   3.3. Familiarización con el entorno de programación de Colab.
   3.4. Análisis de las métricas de evaluación y su implementación.
4. Desarrollo de programas.
   4.1. Implementación de programa para la métrica BLEU.
   4.2. Implementación de programa para evaluación manual.
   4.3. Implementación de programa para las métricas de Sentence Transformers.
   4.4. Desarrollo de programa para el entrenamiento de regresión lineal.
   4.5. Implementación de programa para procesado de resultados.
5. Obtención de resultados y ajustes.
   5.1. Obtención de valoraciones con la métrica BLEU.
   5.2. Obtención de valoraciones con los modelos de Sentence Transformers.
   5.3. Ajuste de resultados del modelo seleccionado de Sentence Transformers.
6. Análisis de los resultados.
   6.1. Análisis de intervenciones aleatorias en distintos grupos según valoraciones.
   6.2. Análisis de la base de datos original.
   6.3. Análisis de las valoraciones de BLEU.
   6.4. Análisis de las valoraciones de Sentence Transformers.

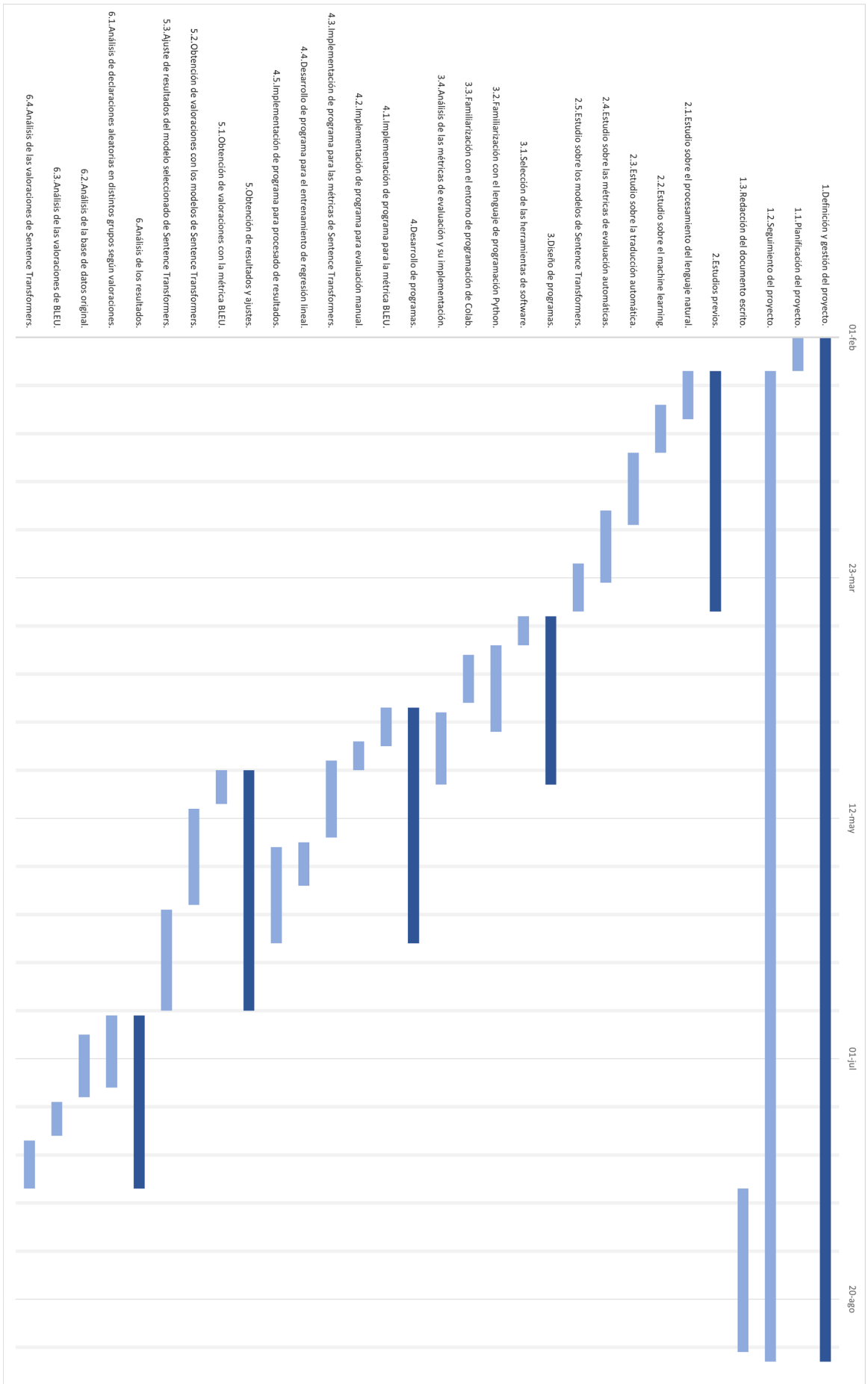
## 1.2 Planificación temporal

La propuesta de trabajo se aceptó en septiembre de 2020, pero se comenzó a desarrollar el 1 de febrero de 2021, dándose por finalizado el 31 de agosto de 2021. Se estima que la duración neta de las actividades sea de unas 390 horas.

La programación temporal del proyecto se presenta en un diagrama de Gantt, donde se asigna a cada uno de los paquetes de trabajo un tiempo estimado de duración dentro del trabajo.

El diagrama, permite visualizar de manera cómoda y representativa el progreso del trabajo en cada una de las etapas que lo conforman.

Durante los primeros meses del proyecto, se centró la atención en la familiarización con el tema de estudio, las herramientas de trabajo y el lenguaje de programación Python. Posteriormente, se realizaron los experimentos pertinentes y el análisis de los resultados obtenidos y se dedicó el último mes a la escritura del presente documento.


1.Definición y gestión del proyecto.
1.1.Planificación del proyecto.
1.2.Seguimiento del proyecto.
1.3.Redacción del documento escrito.
2.Estudios previos.
2.1.Estudio sobre el procesamiento del lenguaje natural.
2.2.Estudio sobre el machine learning.
2.3.Estudio sobre la traducción automática.
2.4.Estudio sobre las métricas de evaluación automáticas.
2.5.Estudio sobre los modelos de Sentence Transformers.
3.Diseño de programas.
3.1.Selección de las herramientas de software.
3.2.Familiarización con el lenguaje de programación Python.
3.3.Familiarización con el entorno de programación de Colab.
3.4.Análisis de las métricas de evaluación y su implementación.
4.Desarrollo de programas.
4.1.Implementación de programa para la métrica BLEU.
4.2.Implementación de programa para evaluación manual.
4.3.Implementación de programa para las métricas de Sentence Transformers.
4.4.Desarrollo de programa para el entrenamiento de regresión lineal.
4.5.Implementación de programa para procesado de resultados.
5.Obtención de resultados y ajustes.
5.1.Obtención de valoraciones con la métrica BLEU.
5.2.Obtención de valoraciones con los modelos de Sentence Transformers.
5.3.Ajuste de resultados del modelo seleccionado de Sentence Transformers.
6.Análisis de los resultados.
6.1.Análisis de declaraciones aleatorias en distintos grupos según valoraciones.
6.2.Análisis de la base de datos original.
6.3.Análisis de las valoraciones de BLEU.
6.4.Análisis de las valoraciones de Sentence Transformers.
01-feb
23-mar
12-may
01-jul
20-ago

## 1.3 Presupuesto

El cálculo del presupuesto se ha estimado según los costes aproximados que supondría el proyecto. Se dividirá el presupuesto en varios grupos:

- Personal
- Material
- Software

### 1.3.1 Personal

Para los costes de personal, se consideran las horas-hombre empleadas por cada uno de los participantes en el proyecto. En este proyecto han participado el tutor del proyecto (profesor contratado doctor), el cotutor (ayudante) y el alumno (ingeniero junior). El coste por hora se ha calculado según la tabla de retribuciones PDI contratados que se puede encontrar en el portal de transparencia de la UPM[33] . El sueldo mensual es a jornada completa, que se considera de 40 horas semanales, por lo tanto, se calcula el coste unitario dividiendo el sueldo por 160 horas mensuales.

| Costes directos | Sueldo mensual (€) | Coste unitario (€/h) | Horas invertidas (h) | Total (€) |
|---|---|---|---|---|
| **Catedrático** | 2.332,44 | 14,57 | 20 | 291,4 |
| **Ayudante** | 1.763,99 | 11,02 | 50 | 551,24 |
| **Alumno** | 1000 | 6,25 | 390 | 2.437,5 |
| **TOTAL** | | | | **3.280,14** |

**TABLA 40. Presupuesto de personal.**

### 1.3.2 Material

El hardware utilizado para el proyecto consta de un ordenador portátil HP Pavilion x360 con procesador Intel i7 de séptima generación, CPU de 2,7 GHz y 12 GB de memoria RAM, con un valor de 1200 €.

| Material | Unidades o meses de uso (#) o (meses) | Precio unitario o precio mensual(€) o (€/mes) | Total (€) |
|---|---|---|---|
| **Hardware** | 1 | 1200 | 1200 |
| **TOTAL** | | | **1.200** |

**TABLA 41. Presupuesto del material.**

[33] https://transparencia.upm.es/

### 1.3.3 Costes indirectos

Los costes indirectos en este caso son el coste de internet con un valor estimado de 210 € (30 € al mes) y el coste del uso de la red eléctrica con un valor estimado de 61 € (8,71 € al mes).

| Costes indirectos | Unidades o meses de uso (#) o (meses) | Precio unitario o precio mensual(€) o (€/mes) | Total (€) |
|---|---|---|---|
| **Red eléctrica** | 7 | 8,71 | 61 |
| **Internet** | 7 | 30 | 210 |
| **TOTAL** | | | **271** |

**TABLA 42. Propiedades de los costes indirectos.**

### 1.3.4 Software

En este proyecto, se ha hecho uso de la herramienta de traducción Azure Translator. El precio de Azure se mide en millones de caracteres y depende del servicio que se escoja. En este caso, se ha escogido el servicio S1 de 10 $ por millón de caracteres de traducción estándar.

| Software | Precio por millón de caracteres ($/#) | Caracteres traducidos (#) | Total ($) | Cambio dólar a euro | Total (€) |
|---|---|---|---|---|---|
| **Azure Translator (Inglés – español) DD** | 10 | 6380264 | 63.80 | 0,847 | 54,03 |
| **Azure Translator (Español- Inglés) DD** | 10 | 6380264 | 63.80 | 0,847 | 54,03 |
| **Azure Translator (Inglés – español) ED** | 10 | 2239346 | 22.39 | 0,847 | 18,96 |
| **Azure Translator (Español- Inglés) ED** | 10 | 2239346 | 22.39 | 0,847 | 18,96 |
| **TOTAL** | | | **127.6** | **0,847**[34] | **145,98** |

**TABLA 43. Presupuesto de software.**

[34] Cambio de un dólar a euro el día 30/08/2021 según https://www.expansion.com/ahorro/conversor-divisas/dolar-euro

### 1.3.4 Resumen de costes

En total, sumando todos los costes se obtiene el siguiente presupuesto final:

| Tipo de coste | Coste (€) |
|---|---|
| **Costes directos** | 3.280,14 |
| **Material** | 1.200 |
| **Costes indirectos** | 271 |
| **Software** | 145,98 |
| **TOTAL** | **4.897,12** |

**TABLA 44. Presupuesto total.**

# ANEXOS

# Índice de figuras

# Índice de tablas

# Bibliografía

[1] Hockett CF, Hockett CD. The origin of speech. Sci Am. 1960;203(3):88-97.

[2] Khurana D, Koli A, Khatter K, Singh S. Natural Language Processing: State of The Art, Current Trends and Challenges. . 2017 /08/17.

[3] Hutchins WJ. Machine translation: past, present, future. Chichester: Ellis Horwood; 1986.

[4] Beck C. The universal character, by which all Nations in the World may understand one another's Conceptions, Reading out of one Common Writing their own Mother Tongues. An Invention of General Use, the Practise whereof may be Attained in two Hours' space, Observing the Grammatical Directions. Which Character is so contrived, that it may be Spoken as well as Written". Londres: Tho. Maxey; 1657.

[5] Turing AM. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London mathematical society. 1937;2(1):230-65.

[6] Turing A. Lecture to the London Mathematieal Society. ; Feb 20, 1947.

[7] Vetulani Z. Human Language Technologies: Tradition and New Challenges.„. Proceedings of Artificial Intelligence. 2005;2(25):5-31.

[8] Richens RH, Booth AD. Some methods of mechanized translation. Machine translation of languages: fourteen essays. 1955:24-46

[9] Weaver W. The mathematics of communication. Sci Am. 1949;181(1):11-5.

[10] Bar-Hillel Y. The present state of research on mechanical translation. American Documentation. 1951;2(4):229-37.

[11] Hutchins WJ. The Georgetown-IBM experiment demonstrated in January 1954. Conference of the Association for Machine Translation in the Americas; Springer; 2004.

[12] Bar-Hillel Y. The present status of automatic translation of languages. Advances in computers. 1960;1:91-163.

[13] Language and machines: computers in translation and linguistics . Washington D.C.: 1966.

[14] Hutchins J. Research methods and system designs in machine translation: a ten-year review, 1984-1994. Citeseer; 1994.

[15] Villayandre Llamazares M. Aproximación a la lingüística computacional. . 2010.

[16] Kapitanova K, Son S. Machine learning basics. In: Intelligent Sensor Networks. CRC Press; 2012. p. 3-29.

[17] Melnikov AV, Botov DS, Klenin JD. On usage of machine learning for natural language processing tasks as illustrated by educational content mining. Онтология проектирования. 2017;7(1 (23)).

[18] De Saussure F, Bally C, Sechehaye A, Riedlinger A, Alonso A, Sechehaye A. Curso de lingüística general. . 1987.

[19] Coseriu E, Polo J. Introducción a la lingüística. Gredos; 1986.

[20] Fernández Pérez M. Las disciplinas lingüísticas. . 1986.

[21] Hockett CF, Hockett CD. The origin of speech. Sci Am. 1960;203(3):88-97.

[22] Han L. Machine translation evaluation resources and methods: A survey. arXiv preprint arXiv:1605.04515. 2016.

[23] Wong B, Kit C. ATEC: automatic evaluation of machine translation via word choice and word order. Machine Translation. 2009;23(2-3):141-55.

[24] Papineni K, Roukos S, Ward T, Zhu W. Bleu: a method for automatic evaluation of machine translation. Proceedings of the 40th annual meeting of the Association for Computational Linguistics; ; 2002.

[25] Barrault L, Bojar O, Costa-Jussa MR, Federmann C, Fishel M, Graham Y, et al. Findings of the 2019 conference on machine translation (wmt19). Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1); ; 2019.

[26] Huang M, Zhu X, Gao J. Challenges in building intelligent open-domain dialog systems. ACM Transactions on Information Systems (TOIS). 2020;38(3):1-32.

[27] Li Y, Su H, Shen X, Li W, Cao Z, Niu S. Dailydialog: A manually labelled multi-turn dialogue dataset. arXiv preprint arXiv:1710.03957. 2017.

[28] Rashkin H, Smith EM, Li M, Boureau Y. Towards empathetic open-domain conversation models: A new benchmark and dataset. arXiv preprint arXiv:1811.00207. 2018.

[29] Li J, Galley M, Brockett C, Spithourakis GP, Gao J, Dolan B. A persona-based neural conversation model. arXiv preprint arXiv:1603.06155. 2016.

[30] Dinan E, Roller S, Shuster K, Fan A, Auli M, Weston J. Wizard of wikipedia: Knowledge-powered conversational agents. arXiv preprint arXiv:1811.01241. 2018.

[31] Shuster K, Urbanek J, Dinan E, Szlam A, Weston J. Deploying lifelong open-domain dialogue learning. arXiv preprint arXiv:2008.08076. 2020.

[32] Ekman P. An argument for basic emotions. Cognition & emotion. 1992;6(3-4):169-200.

[33] Miller AH, Feng W, Fisch A, Lu J, Batra D, Bordes A, et al. Parlai: A dialog research software platform. arXiv preprint arXiv:1705.06476. 2017.

[34] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084. 2019.

[35] Devlin J, Chang M, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018.

[36] Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation. arXiv preprint arXiv:2004.09813. 2020.

[37] Leusch G, Ueffing N, Vilar D, Ney H. Preprocessing and normalization for automatic evaluation of machine translation. Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization; ; 2005.

[38] Pouget-Abadie J, Bahdanau D, Van Merrienboer B, Cho K, Bengio Y. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. arXiv preprint arXiv:1409.1257. 2014.

[39] Stanojević M, Sima'an K. Evaluating MT systems with BEER. The Prague Bulletin of Mathematical Linguistics. 2015;104:17-26.

[40] Siddhant A, Hu J, Johnson M, Firat O, Ruder S. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. . 2020.

[41] Rei R, Stewart C, Farinha AC, Lavie A. COMET: A neural framework for MT evaluation. arXiv preprint arXiv:2009.09025. 2020.

[42] Mathur N, Baldwin T, Cohn T. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. arXiv preprint arXiv:2006.06264. 2020.

[43] Wieting J, Berg-Kirkpatrick T, Gimpel K, Neubig G. Beyond BLEU: training neural machine translation with semantic similarity. arXiv preprint arXiv:1909.06694. 2019.

[44] Reiter E. A structured review of the validity of BLEU. Computational Linguistics. 2018;44(3):393-401.

[45] Koehn P, Knowles R. Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872. 2017.

[46] Scarlini B, Pasini T, Navigli R. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); ; 2020.

[47] Canete J, Chaperon G, Fuentes R, Pérez J. Spanish pre-trained bert model and evaluation data. Pml4dc at iclr. 2020;2020.

[48] Mounin G. La Machine à traduire. La Haye: Mouton. 1964.