
UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de Ingeniería de Sistemas

# Construcción de un Corpus de Gran Escala en el Idioma Español cuyos Documentos Reflejen Opiniones Respecto a Productos Textiles

Trabajo de titulación previo a la obtención del título de Ingeniero de Sistemas

Autor:

David Enrique Santos León

CI: 0104997218

Correo electrónico: david.santos1687@gmail.com

Director:

Ing. Andrés Vinicio Auquilla Sangolquí

CI: 0103557369

**Cuenca, Ecuador**

08-noviembre-2021

# Resumen

Actualmente, existe un auge en introducir modelos de Aprendizaje Automático a varios aspectos de la vida cotidiana. Un campo de relevancia consiste en el Procesamiento del Lenguaje Natural (NLP) que busca modelar al lenguaje humano. La dificultad de entrenar a modelos que aprendan del lenguaje, es alta. Un componente clave y básico para que estas inteligencias aprendan de forma adecuada consiste en los datos, que para el caso de NLP, se encuentran mayoritariamente en inglés. El presente proyecto de investigación surge de la problemática de encontrar insumos de gran escala, en idiomas diferentes al inglés, para alimentar a modelos de Aprendizaje Profundo que produzcan textos de forma automática. Se han generado cuatro resultados principales: 1) Una metodología para construir corpus de gran escala, con facilidad de escalar a diferentes dominios e idiomas, 2) Un corpus en español, dentro del dominio de comentarios de productos textiles, con más de 170 mil documentos que obtuvo buenos resultados de evaluaciones humanas y automáticas, 3) Un sistema computacional que automatizó la construcción del corpus desde el principio al fin, desde la recolección de los documentos hasta su evaluación, y 4) resultados de línea base de un modelo generacional que sirven como punto de referencia para futuras investigaciones dentro de la generación automática de textos dentro del dominio textil.

**Palabras clave:** Construcción de corpus. Corpus en español. Opiniones de productos textiles. Insumos para NLP. Metodología de construcción.

# Abstract

Currently, there is a boom in introducing Machine Learning models to various aspects of everyday life. A relevant field consists of Natural Language Processing (NLP) that seeks to model human language. The difficulty of training models to learn a language is high. A key and basic component for these intelligences to learn properly consists of the data, which in the case of NLP, is mostly in English. This research project arises from the problem of finding large-scale inputs, in languages other than English, to feed Deep Learning models that produce texts automatically. Four main results have been generated: 1) A methodology to build a large-scale corpus, easily scalable to different domains and languages, 2) A corpus in Spanish, within the domain of comments on textile products, with more than 170 thousand documents that obtained good results from human and automatic evaluations, 3) A computational system that automated the construction of the corpus from beginning to end, from the collection of documents to their evaluation, and 4) baseline results of a generational model that serve as a point of reference for future research within the automatic generation of texts within the textile domain.

**Keywords:** Construction methodology. Corpus construction. Corpus in spanish. Supplies for NLP. Textile product reviews.

# Índice general

# Índice de figuras

# Índice de tablas

# Cláusula de Propiedad Intelectual

David Enrique Santos León, autor del trabajo de titulación Construcción de un Corpus de Gran Escala en el Idioma Español cuyos Documentos Reflejen Opiniones Respecto a Productos Textiles , certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 8 de noviembre de 2021

David Santos

David Enrique Santos León

C.I: 0104997218

# Cláusula de licencia y autorización para publicación en el Repositorio Institucional

David Enrique Santos León en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación Construcción de un Corpus de Gran Escala en el Idioma Español cuyos Documentos Reflejen Opiniones Respecto a Productos Textiles de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 8 de noviembre de 2021

David Santos

David Enrique Santos León

C.I: 0104997218

# Dedicatoria

Para mi esposa e hija: Liz y Raffa.

**David Santos**

# Agradecimientos

Agradezco a mi familia, pilar durante este proceso. Cada uno tuvo un momento en el que pudo ser el apoyo que necesitaba para llegar a este punto: mis papás, mis hermanos, y mi suegro. Esto no hubiera sido posible sin ellos. Gracias por creen en mi, y en que esta era la decisión correcta. A mi esposa e hija agradezco su apoyo y compañía en caminos que al principio estaban llenos de incertidumbre.

Agradezco al Ing. Andrés Auquilla por haber dirigido este proyecto, y por haber sido guía para una proyección profesional a futuro.

Finalmente, agradezco al proyecto *Incorporationg Sustainability concepts to management models of textile Micro, Small and Medium Enterprises (SUMA)*, junto a todos los colegas que ahí desempeñan funciones. De un trabajo conjunto fue concebida la idea del presente proyecto de titulación. Además, lograron gestionar recursos para acceder a tecnologías de altas prestaciones.

**David Santos**

# Abreviaciones y Acrónimos

**APIs** Application Programming Interfaces. 5

**SDK** Software Development Kit. 6

**URLs** Uniform Resource Locators. 31

**UTF-8** 8-bit Unicode Transformation Format. 7

Capítulo 1

# Introducción

"Mientras examinaban a estas bizarras criaturas los científicos descubrieron que las criaturas también hablaban un inglés regular (...)" (OpenAI, 2019). El anterior pasaje refiere al descubrimiento de unicornios por parte de científicos, ¡Unicornios que hablan inglés! Rápidamente se deduce que el texto contiene una falacia; sin embargo, lo que no se puede deducir con facilidad es que el texto, que describe acontecimientos, introduce personajes y hasta cita textualmente palabras de los personajes, fue creado por un modelo de Inteligencia Artificial (AI por sus siglas en inglés). La tarea de generar textos inteligibles se encuentra dentro del campo del Procesamiento del Lenguaje Natural (NLP por sus siglas en inglés), una disciplina que afronta uno de los problemas más antiguos y difíciles de la AI (Vieira y Ribeiro, 2018): el modelado del lenguaje humano.

## 1.1. Identificación del problema

Investigaciones realizadas en esta área han logrado resultados prometedores en los últimos años: generación de artículos descriptivos completos (Brown et al., 2020a), oraciones dentro de categorías definidas (Li et al., 2018a), conjuntos de oraciones resultado de interpolar entre dos oraciones (Bowman et al., 2015b) e incluso oraciones controlando sentimientos, tono, tiempo, voz y humor (Logeswaran et al., 2018). A pesar de utilizar enfoques y arquitecturas diferentes estas investigaciones tienen dos factores en común: 1) utilizan técnicas de Aprendizaje Profundo (AP), un sub campo de AI en donde a través de redes neuronales organizadas en múltiples capas se pueden abordar tareas complejas (Chassagnon et al., 2020), y 2) estos algoritmos son entrenados con documentos en inglés. Estos factores se encuentran estrechamente conectados; los algoritmos de AP requieren cantidades ingentes de datos (Ng et al., 2017) y los corpus que se definen como grandes cuerpos de evidencia lingüística compuestos por el uso del lenguaje atestiguado (McEnery, 2012), se encuentran principalmente en idioma inglés (Ray et al., 2019).

Esta tendencia hacia los insumos en inglés se convierte en un impedimento para los investigadores que buscan reproducir los resultados de los modelos generacionales de vanguardia en otros idiomas (Conneau et al., 2018). Keung et al. (2020) mencionan que la existencia de corpus de gran escala en otros idiomas es extremadamente rara y estos por lo general presentan deficiencias. El corpus Yelp de uso no comercial (Yelp, 2019) posee más de 8 millones de registros en varias lenguas, dentro del

dominio de comentarios sobre servicios de comida, pero no cuenta con identificadores para el lenguaje de los documentos. El corpus RVC2 generado por Reuters (Reuters, 2019) se conforma por reportajes producidos por sus corresponsales; cuenta 487 mil documentos divididos en 14 idiomas (incluido el español), sin embargo, el acceso a estos datos es restringido debiendo cumplir con varios requisitos para obtener el link de descarga, como por ejemplo pertenecer a una institución que mantenga un convenio con REUTERS y que demuestre que sus investigaciones giren en torno al NLP. El corpus de Amazon Review (McAuley, 2018) dispone de comentarios en varios idiomas pero estos no han pasado por una fase de limpieza.

## 1.2. Justificación

Para reducir la complejidad de la adaptación de técnicas para generación de texto al idioma español, el presente trabajo explora las posibilidades de construir corpus en español a través de: 1) la recuperación de documentos en la red social Twitter por medio de su API oficial, 2) la adaptación de corpus públicos en idioma español, y 3) la adaptación de corpus públicos con documentos en inglés para ser utilizados en español por medio traductores automáticos. Los resultados principales a obtener consisten en un levantamiento de información acerca de corpus disponibles, en una metodología para generar y evaluar corpus en español, un corpus de gran escala con capacidad de alimentar a modelos de NLP en idioma español, y resultados de línea base para la generación automática de textos.

La información utilizada para el caso de estudio corresponde a comentarios sobre productos textiles. Esto debido a que esta industria dentro del Ecuador genera el 21 % de las plazas totales de trabajo (Gómez, 2021); y, además de enfrentarse a factores como competencia desleal y contrabando (Ordóñez, 2015), ha sido duramente golpeada por la pandemia, registrando en el 2020 pérdidas de $ 500 millones, un decremento del 36 % con respecto al año anterior Angulo (2021). El presente trabajo de investigación se desarrolla dentro del proyecto ***Incorporating Sustainability Concepts to Management Models of Textile Micro, Small and Medium Enterprises*** **(SUMA)**, cuyo objetivo es "Implementar Modelos de gestión sostenibles en MIPYMES para optimizar su desempeño y sentar la base para una economía verde y de mayor sostenibilidad, específicamente dentro del sector textil" (Universidad de Cuenca, 2020).

Los resultados teóricos y prácticos obtenidos por este trabajo tienen un impacto positivo en la industria textil nacional. Mediante la metodología e insumos obtenidos, las empresas podrán disponer las bases para generar productos como contenido publicitario automatizado destinado a ser desplegado en campañas de marketing, o modelos que permitan analizar el sentimiento de sus clientes hacia sus productos. Por otro lado, el aporte teórico y los insumos obtenidos ayudarán a reducir la brecha del estado del arte en este tema, ya que la metodología propuesta es fácilmente adaptable para desarrollar trabajos futuros de NLP dentro del idioma español.

## 1.3. Objetivos

### 1.3.1. Objetivo general

Generar un corpus de gran escala en español mediante un proceso estandarizado y escalable que tenga en cuenta la calidad de los documentos, y que sea capaz de alimentar modelos de Procesamiento de Lenguaje Natural, cuyos documentos posean una semántica que gire en torno a opiniones respecto a productos textiles.

### 1.3.2. Objetivos específicos

El presente trabajo tiene los siguientes objetivos específicos:

- Identificar y caracterizar a los principales corpus en inglés y español, definiendo además las características que debe poseer el corpus para ser alimentado a modelos generacionales que automaticen la generación de textos.
- Definir la metodología para la recolección, procesamiento, limpieza, y almacenamiento de los datos que crearán el corpus en español junto con el desarrollo de los sistemas que se encargarán de esta tarea.
- Construir un corpus de acuerdo a las caracterizaciones y metodologías descritas.
- Evaluar el corpus generado junto con los resultados de línea base de un modelo generacional.

## 1.4. Estructura del documento de investigación

El presente documento de investigación se estructura en seis capítulos:

1. **Introducción.** Se introduce el proyecto de investigación, se presenta el planteamiento del problema, la justificación, los objetivos y la estructura del documento.
2. **Marco Teórico.** Se presentan los conceptos teóricos básicos para la comprensión de los temas que se desarrollan.
3. **Estado del Arte y Trabajos Relacionados.** Se referencian los trabajos actuales que se desarrollan para la construcción de corpus así como la generación automática de textos; se indican también trabajos relevantes al área.
4. **Diseño e Implementación.** Se diseña la metodología a utilizar para construir el corpus. Se diseñan y construyen los sistemas que se encargarán de recolectar los documentos. Finalmente, se pone en marcha la propuesta a través de su implementación.
5. **Resultados.** Se describen los resultados principales relacionados a la sistematización de corpus disponibles, la construcción del corpus, la evaluación del corpus, y los resultados de línea base del modelo generacional.
6. **Conclusiones y Recomendaciones.** Se presentan las conclusiones del proyecto, se proporcionan recomendaciones así como futuras líneas de trabajo para extender a esta investigación.

Capítulo 2

# Marco Teórico

El presente capítulo describe los conceptos básicos para la comprensión de los temas que se desarrollan dentro de la investigación. Describe las características de un corpus, presenta algunas fuentes para obtener documentos, describe medios y formatos para la distribución de corpus, presenta plataformas y herramientas para realizar traducciones lingüísticas, describe métricas automáticas y manuales para evaluar la calidad de documentos generados por una máquina, presenta los tipos de licencia comunes para la distribución de corpus y finalmente introduce a conceptos básicos de Aprendizaje Automático (AA) junto con tecnologías que dan soporte a la implementación de estas tareas.

## 2.1. Caracterización de un Corpus

Existen varias definiciones de un corpus, McEnery (2012) lo denominan como grandes cuerpos de evidencia lingüística compuestos por el uso del lenguaje atestiguado tanto escrito como hablado. Por su parte, Liu y Han (2012) lo definen como un conjunto grande y estructurado de textos, pudiendo ser de un solo lenguaje (monolingüe) o de varios (multilingüe); dentro de los corpus multilingües existen los denominados corpus paralelos que contiene documentos en un idioma con sus correspondientes traducciones en otros lenguajes (Miangah, 2009). Su objetivo es servir como herramienta o insumo que permita conocer varios aspectos del lenguaje que de otra manera permanecerían ocultos (O'Keeffe y McCarthy, 2010). Existen de dominio general y específico; dentro del campo de la IA, los modelos que presentan mejores resultados en cuanto al modelamiento del lenguaje están altamente restringidos a un dominio específico (Bengfort et al., 2018).

La elaboración de estos insumos requiere de un trabajo metodológico y muchas horas de preparación; un corpus de alto valor no es una colección aleatoria de documentos, debiendo cumplir con características de: tipo, tamaño, y composición para una aplicación concreta. El tamaño del corpus debe estar en concordancia con los objetivos, tareas y restricciones de la investigación. Corpus especializados pequeños tienen un alto valor para explorar características gramaticales comunes, en cambio los corpus de gran escala son útiles para el análisis de características gramaticales extrañas (O'Keeffe y McCarthy, 2010).

Los corpus pueden ser anotados (junto con los documentos se agregan anotaciones por parte de los investigadores que describen dimensiones adicionales a partir de los atributos originales), o no anotados

(Bengfort et al., 2018). Liang et al. (2020) agrupan a las tareas relacionadas al campo de NLP que se realizan sobre corpus en tres categorías principales:

**Tareas de entendimiento de una sola entrada.** Tareas que buscan generar aprendizaje a través de una sola entrada, e.g., tareas de clasificación de textos, part-of-speech (POS) tagging (proceso de asignar a cada palabra su etiqueta referente a la parte del discurso (Bengfort et al., 2018), y la identificación de entidades nombradas dentro de un texto.

**Tareas de entendimiento de entrada par.** A diferencia de la categoría anterior, cada entrada consiste en un par de documentos. Aquí destacan las tareas de sumarización, preguntas y respuestas, inferencia entre causalidad y contradicción, y, ranking de sitios web para una consulta entregada.

**Tareas de generación.** Buscan generar textos de forma automática con características similares a las de los corpus de entrenamiento. Dentro de esta categoría se mencionan la generación de pasajes, preguntas, títulos de noticias entre otros.

## 2.2. Plataformas, Sitios y Herramientas para Recolectar Documentos en Internet

Los investigadores en las últimas décadas han basado la construcción de corpus a partir de documentos obtenidos en Internet. Para esto se han valido de herramientas para recorrer y recolectar datos en sitios web, Application Programming Interfaces (APIs) oficiales, y corpus distribuidos a través de repositorios. A continuación se detallan estas fuentes de datos.

### 2.2.1. Web Scraping

Consiste en la técnica automatizada para recolectar datos a través de Internet; usando cualquier medio diferente (y automatizado) a interactuar con una API (Mitchel, 2018). Un *web crawler* es definido como un programa o sistema que recorre a través de varios sitios web descargando información de manera sistémica (AbuKausar et al., 2013). Estos sistemas pueden ser desarrollados por los investigadores para cumplir con una tarea específica, o también adquiridos a empresas que se encargan de desarrollar versiones más genéricas proporcionando licencias gratuitas y de pago.

Esta técnica no consiste en la panacea para recuperar datos de sitios web. Requiere alta especialización para una tarea concreta; no existe un estándar para construir un sitio web. Mitchel (2018) indica que antes de aplicar esta técnica se debe realizar un estudio previo al sitio web objetivo en donde se identifique el modo en el que se encuentra organizado, esto se traduce en una dificultad para escalar al sistema. Adicionalmente, existen plataformas como *Facebook* que implementan mecanismos para evitar el ingreso de *web crawlers*, y además son capaces de emprender acciones legales contra investigadores que busquen recolectar información a través de métodos automáticos dentro de su dominio (Mancosu y Vegetti, 2020).

### 2.2.2. APIs Oficiales de Redes Sociales

Una API consiste en la interfaz que los sistemas computacionales presentan a otros sistemas, humanos, y en el caso de APIs web, al mundo a través de Internet (Jin et al., 2018). Las APIs que proveen algunas redes sociales proveen una forma escalable y sencilla para la creación de fuentes de datos; estas poseen diferentes niveles de acceso a los datos, así como restricciones a los mismos. Entre

las principales APIs para obtener documentos de texto se mencionan las que se encuentran dentro de los 10 principales sitios web donde los usuarios comparten sus opiniones sobre productos comprados en Internet a nivel mundial (Statista, 2016):

**Twitter.** Si bien consiste en la API que proporciona la mayor cantidad de datos, la información de las cuentas públicas no es totalmente disponible; Twitter posee políticas de privacidad que impiden la divulgación información como correos, nombres, apellidos, y teléfonos, además de datos que no hayan sido proporcionados dentro del formulario de registro como por ejemplo ubicaciones. Usuarios con permisos de desarrollador pueden obtener datos de cuentas públicas a través conexiones agrupadas en cinco categorías (Twitter, 2021): 1) cuentas y usuarios, 2) tuits y respuestas, 3) Mensajes directos, 4) Anuncios y 5) Herramientas y Software Development Kit (SDK) del editor. Adicionalmente, existen cuotas de uso, restringiendo el acceso a un número de solicitudes al servidor y al número de documentos recuperados en un intervalo de tiempo. Se pueden acceder a planes más flexibles para el acceso a los datos bajo condiciones de pago.

**YouTube.** Proporciona una interfaz llamada YouTube Data API, la cual además de permitir la implementación de funcionalidades normalmente ejecutadas dentro de la plataforma en otros sitios web, entrega recursos para acceder a los mensajes que los usuarios publican como comentarios en los videos. El uso de esta API es gratuito y se encuentra sujeto a una cuota de uso de 10 mil unidades (puntos) por día; se busca evitar que los usuarios generen clones de la plataforma (YouTube, 2021). Cada tipo de servicio consume un número de puntos definido, en el caso de los mensajes de texto se indica que toma un punto de la cuota disponible el recuperarlo o editarlo. Si bien se puede solicitar una extensión de la cuota, se tiene que pasar por un proceso de validación en donde personal de la plataforma audita la solicitud.

**Facebook.** De acuerdo a Statista (2016), en el año en referencia esta plataforma consistió en la primera red social a nivel mundial donde los usuarios compartieron sus opiniones sobre productos adquiridos. Sin embargo, también es conocida por proporcionar una de las APIs más restringidas (Aswani et al., 2018). Por su parte, *Facebook* indica que buscan proteger la información de sus usuarios mientras permiten a los desarrolladores crear experiencias útiles (Schroepfer, 2018). Para acceder a esta API los desarrolladores deben registrarse en la plataforma y entregar su aplicación a modo de ser auditada. Una vez aprobada, se proporciona un token de acceso, al cual los usuarios de la aplicación deben conceder permiso para que se recolecten sus datos de su cuenta de *Facebook* (Facebook, 2021). Existen otros medios oficiales para acceder a estos datos, así lo demuestran Bhattacharya et al. (2020) quienes a través de un convenio obtuvieron un acceso abierto a los comentarios de los perfiles públicos para modelar comportamientos agresivos y misóginos dentro de plataformas digitales.

### 2.2.3. Corpus de Acceso Abierto

En varias ocasiones corpus desarrollados por organizaciones, investigadores y empresas privadas son liberados al público bajo diferentes tipos de licencias. Muchas investigaciones han hecho uso de estos insumos como punto de partida para extenderlos o modificarlos según los objetivos de la investigación. Dentro del Capítulo 3 se realiza un análisis a mayor profundidad de este tema.

## 2.3. Medios y Formatos de Distribución de un Corpus

Si bien un corpus se puede distribuir por cualquier medio, normalmente requiere de una gran capacidad de almacenamiento, lo cual representa una dificultad; por lo que, los investigadores aprovechan la capacidad en la nube para su almacenamiento y distribución.

### 2.3.1. Medios de Distribución

Las principales al momento de distribuir un corpus son: 1) a través de un sitio web propio y 2) a través de repositorios públicos. En el primer caso, el corpus se encuentra disponible bajo el dominio y hosting de parte de quien lo proporciona. En el segundo caso el corpus se encuentra alojado en servicios de terceros; se han identificado exclusivamente el uso de las plataformas *GitHub* y *Open Data on AWS* para este tipo de almacenamiento. Una gran ventaja de Open Data on AWS consiste en la integración directa del corpus con otros servicios que provee AWS (Amazon, 2021), de esta manera se puede trabajar sobre el corpus sin necesidad de descargarlo.

Aunque GitHub busca proporcionar almacenamiento abundante dentro de su plataforma para cuentas gratuitas, poseen límites para los tamaños de archivos y repositorios (GitHub, 2021). La plataforma recomienda que los repositorios sean de menos de 1 GB (aunque permiten almacenar hasta 5 GB), y tienen un límite de tamaño máximo de 100 MB para cada archivo. En cambio, la plataforma *Open Data on AWS* no menciona una cuota de uso, no obstante, los conjuntos de datos que almacenan son de carácter público y requieren de un registro previo para su aprobación (AWS, 2021).

### 2.3.2. Formatos de los Documentos

Un corpus se encuentra constituido por documentos, pudiendo estos estar conformados por diferentes tipos de unidades de análisis: palabras, oraciones, párrafos, conjuntos de párrafos o incluso conjuntos de documentos. Se ha identificado dentro de la literatura el uso extendido 8-bit Unicode Transformation Format (UTF-8) para codificar a los caracteres, siendo esta verificación incluso parte de la limpieza que se realizan en varias investigaciones.

Se utilizan varios tipos de formatos para almacenar documentos y facilitar su posterior recuperación. Estos formatos conforman un modelo abstracto en particular (Hogan, 2020), y en algunos casos se combinan para aprovechar sus fortalezas (Minard et al., 2016). Los formatos identificados en la literatura referente a la construcción de corpus son los siguientes:

**Comma Separated Values (CSV).** Los valores de los atributos se separan por *comas*, cada registro se localiza en una línea separada, delimitada por el carácter de salto $\backslash n$ (Shafranovich, 2005).

**Texto Delimitado por Tabulaciones (TSV).** Similar a CSV, salvo que en vez del carácter *coma* se utiliza el carácter de *tabulación* (UPAEP, 2021).

**JavaScript Object Notation (JSON).** Formato construido a través de dos estructuras: 1) pares *clave/valor* y 2) una lista de valores asociados separados por *comas*; tiene la ventaja de ser sencillo de leer y escribir, tanto para humanos como para máquinas (ECMA-404, 2021a).

**JSON Lines.** Formato con tres requerimientos: 1) Codificación UTF-8, 2) Cada línea es un valor JSON válido, y 3) El separador de línea consiste en un carácter de salto (ECMA-404, 2021b).

**Extensible Markup Language (XML).** Formato para representar información estructurada, en donde a través de etiquetas se siguen reglas sintácticas que permiten definir estándares (W3C, 2015).

## 2.4. Plataformas y Herramientas para Traducciones Lingüísticas

Existe variedad de herramientas y plataformas para realizar traducciones entre idiomas. A través de la revisión bibliográfica se han encontrado dos categorías que engloban a las herramientas dedicadas a esta tarea: mecanismos manuales y automáticos. A continuación, se presenta una sistematización de plataformas para traducción con capacidades de escalar ante una alta cantidad de documentos a traducir.

### 2.4.1. Mecanismos Manuales

Consisten en traducciones realizadas por personas. Si bien existe una plataforma tecnológica que se encarga de la logística en cuanto a pagos, asignación de recursos, carga de trabajo, etc., al final la traducción la realiza un componente netamente humano.

**Amazon Mechanical Turk (MTurk).** Mercado *crowdsourcing* que facilita subcontratar procesos y tareas a una fuerza de trabajo distribuida que puede realizarlos de manera virtual (MTurk, 2021). En esta plataforma se pueden ofertar todo tipo de tareas siempre que cumplan con los términos de uso de *Amazon.com*, no obstante, existe un factor de incertidumbre en conseguir el personal requerido para la tarea generada ya que pudiera presentarse el escenario en que no existe demanda laboral para la oferta presentada. En esta plataforma el valor de la tarea es fijado por la parte que requiere del servicio.

**Plataformas especializadas en servicios lingüísticos.** Existen plataformas destinadas a proveer servicios de traducción a través de personas especializadas en idiomas. Presentan la ventaja de garantizar profesionales certificados y además se elimina la incertidumbre de conseguir un profesional para una determinada traducción. Un ejemplo de estas plataformas es *One Hour Translation*, que indican contar con una red de 25 mil profesionales brindando soporte a 120 lenguajes (One Hour Translation, 2021). A diferencia de MTurk, los costos son definidos por la parte ofertante del servicio, variando de acuerdo al tipo de lenguaje.

### 2.4.2. Mecanismos Automáticos

Corresponden a medios en los que las traducciones las realizan sistemas informáticos. A continuación, se presentan las opciones consideradas como relevantes para la presente investigación.

**Cloud Translation API.** Plataforma desarrollada por *Google* que permite la traducción a través de modelos de AA (Google Cloud, 2021). En su edición básica permite la traducción de textos a más de 100 idiomas. Adicionalmente, proporciona modelos pre entrenados, que sirven como base para entrenar otros modelos en nuevos idiomas. Aplica tarifas basadas en número de caracteres; proporciona un límite gratuito de 500 mil caracteres al mes, equivalentes a 40 dólares de consumo.

**Translator Text API.** Servicio de IA proporcionado por *Microsoft Azure* para la traducción automática de texto entregando soporte a 90 idiomas (Microsoft Azure, 2021). Otorgan una cuota gratuita en la versión de prueba por 2 millones de caracteres por mes.

**Moses.** Consiste en un sistema de escritorio para la traducción automática estadística a través del entrenamiento de modelos traductores en cualquier par de lenguajes; requiere corpus paralelos (Statistical Machine Translation, 2017). A diferencia de los sistemas traductores mencionados

anteriormente, cuenta con la licencia LGNU, permitiendo utilizar al sistema sin restricciones de cuotas de uso.

## 2.5. Métricas para la Evaluación de Corpus

Siguiendo el esquema realizado por Bowman et al. (2015a) la evaluación de un corpus se puede dividir en dos componentes: medios automáticos y medios humanos. Para los medios automáticos, se hace referencia a las métricas que buscan determinar cuantitativamente la calidad de un conjunto de documentos generados por modelos computacionales. En cambio, para los medios humanos se integra un componente cualitativo sobre la apreciación de las personas.

### 2.5.1. Evaluación por Medios Automáticos

Se introducen las métricas que buscan cuantificar la calidad de un texto generado por medios automáticos. Entre las principales se mencionan:

**Bilingual Evaluation Understudy (BLEU).** Presentada por Papineni et al. (2002) consiste en una métrica para medir la calidad de una traducción por métodos (Ec. 2.1). Tiene como idea central cuantificar que tan similar es una traducción realizada por un modelo frente a la misma traducción realizada por un traductor (persona) profesional. Calcula un factor de precisión que corresponde al número de ocurrencias de *n-gramas* dentro de ambas traducciones (conjunto $C$). Este valor se encuentra en 0 y 1, representando un valor de 1 una coincidencia exacta entre las dos traducciones. Posteriormente, pondera un factor de penalización a la media geométrica.

$$p_n = \frac{\sum_{C \in Candidates} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in Candidates} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')} \tag{2.1}$$

**Metric for Evaluation of Translation with Explicit ORdering (METEOR).** Introducida por Banerjee y Lavie (2005) (Ec. 2.2) consiste en una métrica basada en unigramas para evaluar traducciones automáticas frente a las realizadas por personas. A diferencia de BLEU, METEOR toma en consideración el *precision* y *recall* a través de la media armónica *F-mean* de manera directa. Se denomina a *chunks* como el número de segmentos que se pueden tomar de acuerdo a los n-gramas seleccionados.

$$Score = F_{mean} \cdot (1 - 0{,}5 \cdot \frac{\#chunks}{\#unigrams\_matched}) \tag{2.2}$$

**Distinct-N.** Propuesta por Li et al. (2015) consiste en una métrica que mide la diversidad de una oración generada por un sistema. Analiza el número de *n gramas* diferentes y penaliza a oraciones con varios *n gramas* repetidos. En su forma común se realiza un análisis en unigramas y bigramas, Distinct-1 y Distinct-2 respectivamente.

**Recall-Oriented Understudy for Gisting Evaluation - L (ROUGE-L).** En su versión inicial (ROUGE), presentada por Lin (2004) buscó determinar la calidad de resúmenes generados de forma automática. Posteriormente Lin y Och (2004) extienden el trabajo original, creando ROUGE-L para abarcar el campo de la traducción automática. Esta métrica mide la frecuencia de traslapes de *n gramas* entre un texto generado por un sistema frente a uno considerado como ideal generado por personas.

**NIST.** Variante de BLEU desarrollada por el *National Institute of Standars and Technology* (NIST) (Lin y Och, 2004) que considera información adicional de los *n gramas*. Mientras que

BLEU utiliza la media geométrica dentro de su formulación, NIST utiliza la media aritmética de los *n gramas* tomando en consideración la longitud de los segmentos (Surcin et al., 2005).

### 2.5.2. Evaluación Humana

No existe un estándar con respecto a la evaluación humana de un corpus construido. Por lo general los investigadores complementariamente a las evaluaciones automáticas, generan diseños experimentales para evaluar el contenido a través de criterios humanos según sus necesidades y disponibilidades. Por ejemplo, Fair y Gardent (2020) construyeron una escala de tres puntos para evaluar características morfológicas de las palabras a partir de un cuestionario, mientras que Bowman et al. (2015a) realizaron una tarea de asociación de textos de salida con un conjunto de opciones, entre la que se encuentra la respuesta correcta, logrando determinar un porcentaje de textos correctos. Tomando en cuenta los trabajos de (Ni et al., 2020) y (Fair y Gardent, 2020), se presenta un compendio de las características evaluadas por el componente humano más utilizadas y de relevancia para el presente estudio:

**Relevancia.** Mide si un texto de salida contiene información relevante al tema de estudio.
**Informatividad.** Mide si un texto de salida incluye información relevante a los usuarios.
**Diversidad.** Mide que tan distinto es un texto de salida comparado con otros.
**Fidelidad semántica.** Determina si un texto de salida guarda relación semántica con su versión original.
**Morfología.** Mide el grado en que las palabras de un texto de salida conservan una morfología adecuada de acuerdo a su contexto.

## 2.6. Licencias Comunes para la Distribución de Corpus

El objetivo de agregar una licencia a un corpus consiste en describir claramente el correcto uso y las responsabilidades agregadas para quien lo utilice (Papers With Code, 2021). De esta manera se alienta a los usuarios a tomar decisiones informadas sobre el uso de los documentos y se minimiza el riesgo del uso malicioso que puede desencadenar consecuencias no deseadas. Un mismo corpus puede incluir varios tipos de licencias, de acuerdo a las fuentes de los documentos. A continuación, se presentan los principales tipos de licencias usados en los conjuntos de datos, incluidos corpus, dentro del campo del Aprendizaje Automático (Papers With Code, 2021):

**CC0.** Dominio público. Permite a los creadores renunciar a sus derechos de autor y colocar su creación al dominio público.
**CC BY.** Permite a los usuarios distribuir modificar, adaptar y construir sobre el material siempre que se entregue atribución al creador original. Permite uso comercial.
**CC BY-SA.** Igual que la licencia CC BY, pero toda adaptación o extensión realizada debe conservar la misma licencia del corpus original.
**CC BY-NC.** Similar a la licencia CC BY, no obstante, no permite el uso comercial del corpus.
**CC BY-NC-SA.** Conserva las características de la licencia CC BY-NC, y adicionalmente cualquier cambio o extensión debe ser licenciado bajo la licencia del corpus original.
**CC BY-ND.** Esta licencia permite copiar y distribuir el material por cualquier medio o formato pero sin modificar el contenido original. Se debe entregar atribución al autor.
**CC BY-NC-ND.** Licencia con condiciones de uso similares a la CC BY-ND, pero no permite el uso comercial.

**Apache-2.0.** Licencia perpetua, irrevocable, no exclusiva y sin cargos/comisiones. Permite reproducir el contenido, preparar trabajos derivados, crear sub licencias, y distribuirlo por cualquier medio. Requiere conservar los derechos de autor y descargos de responsabilidad. Para distribuir versiones modificadas no se requiere incluir códigos fuente (Open Source Initiative, 2021a).

**MIT.** Licencia permisiva utilizada principalmente para productos de software libre y sus asociados, pudiendo ser sets de datos. Entrega permiso libre de cargo a cualquier persona que obtenga una copia del producto en referencia, pudiendo esta copiar, modificar, publicar, crear sublicencias y/o vender copias del producto; expresa la no garantía en ningún aspecto (Open Source Initiative, 2021b).

**GNU Lesser General Public License (LGNU).** Permite copiar y distribuir los productos pero si se realizan modificaciones estas deben conservar la misma licencia (GNU, 2021).

**Personalizada.** Licencia no estandarizada que se basa en los términos de uso de la parte que provee el insumo.

**Desconocida.** Refiere a un corpus del que su licencia no se conoce o no ha sido registrada todavía.

## 2.7. Conceptos Básicos sobre Aprendizaje Automático (AA)

El Aprendizaje Automático (AA) es la práctica que trata en ayudar a un software a realizar una tarea sin establecer reglas específicas (TensorFlow, 2021a). Para esto, trabaja a partir de modelos que consisten en funciones que dependen de un conjunto de parámetros, y transforman un vector de entrada $X$ a un vector de salida $Y$ (Bonaccorso et al., 2018); el vector de entrada debe provenir de un muestreo de valores independientes e idénticamente distribuidos (i.i.d).

Parte de la metodología de trabajo consiste en dividir al conjunto inicial de datos en tres subconjuntos (Brownlee, 2017): 1) Set de entrenamiento: para entrenar al modelo y 2) Set de validación para evaluar métricas del modelo con ejemplos no vistos antes mientras se ajustan los hiper parámetros del modelo, y 3) Set de evaluación, que entrega una evaluación final del modelo sobre datos no vistos antes.

Durante la fase de entrenamiento, los modelos buscan optimizar una función objetivo. Para el presente estudio esta función consiste en *Sparse Categorical Crossentropy* (Ec. 2.3), la cual es la extención de *Categorical Crossentropy* y se utiliza cuando existen dos o más etiquetas de clase (TensorFlow, 2021b); en donde $M$ es el número de ejemplos, $y$ es la etiqueta de clase real y $p$ la etiqueta de clase predecida.

$$SparseCrossentropy = -\sum_{c=1}^{M} y_{o,c} \cdot log(p_{o,c}) \tag{2.3}$$

### 2.7.1. Modelos de Aprendizaje Automático (AA)

No es posible determinar la cantidad de modelos que existen debido a la rapidez con que se desarrollan nuevos algoritmos, aunque sí se pueden definir las categorías que los engloban de acuerdo al tipo de aprendizaje. Brownlee (2019) expone 14 tipos de aprendizaje enmarcados en cuatro tipos de problemas: problemas de aprendizaje, problemas híbridos, problemas de inferencia estadística y técnicas de aprendizaje. En la presente investigación se trabajará dentro del primer tipo de problema que abarca los aprendizajes: supervisado, no supervisado, y por refuerzo.

### 2.7.2. Aprendizaje Supervisado

Consiste en aprender una función que mapee un conjunto de variables de entrada hacia un conjunto de variables de salida; de modo que, la función aprendida pueda ser utilizada para predecir las salidas para las entradas no vistas con anterioridad por el modelo (Cunningham et al., 2008).

En este tipo de aprendizaje los modelos trabajan sobre una colección de datos etiquetados que pueden ser representados por tuplas $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, en donde el vector $x_i$ corresponde a la entrada, y el valor $y_i$ corresponde a la salida. Aunque típicamente el modelo que se construye es una representación de una función matemática, resulta conveniente apreciarlo como un modelo que observa datos de entrada y basado en su experiencia con ejemplos pasados, entrega una salida (Burkov, 2020).

### 2.7.3. Aprendizaje Profundo (AP)

Un tipo de modelo basado en redes neuronales emerge en la industria a fines de los años 90, aunque sus conceptos básicos inician su desarrollo desde la década de los 40. Esto debido principalmente a los desarrollos de hardware que incrementaron notablemente la capacidad computacional (Van Der Smagt y Krose, 1996).

La técnica de modelado por redes neuronales se inspira en la estructura y funcionamiento del cerebro. Una red puede ser dividida en tres partes conocidas como capas (Karsoliya, 2012): 1) Capa de entrada, 2) Capa oculta, y 3) Capa de salida. Existen varios tipos de arquitecturas y topologías disponibles, entre las que se pueden mencionar (Bonaccorso et al., 2018) y (Da Silva et al., 2017):

**Feedforward de una sola capa.** Red con una sola capa de entrada y una sola capa neuronal que consiste en la salida (Figura 2.1). La información fluye en sentido unidireccional. Dentro de esta categoría destacan los tipos de redes Perceptrón y ADALINE.

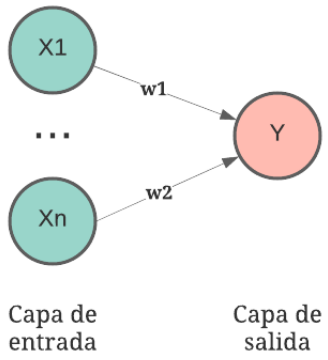


Figura 2.1: Feedforward de una sola capa.

**Feedforward de múltiples capas.** Difieren de las anteriores en que poseen múltiples capas neuronales entre la capa de entrada y de salida (Figura 2.2). Aquí destaca el Perceptrón Multicapa (MLP por sus siglas en inglés).

**Redes Neuronales Recurrentes (RNN por sus siglas en inglés).** En estas redes la salida de una capa de neuronas es utilizada como entrada de realimentación para otras capas; de este modo, se genera un historial de las entradas. En este tipo de redes sobresalen los modelos Long-Short-Term Memory (LSTM) y Gate Recurrent Unit (GRU). LSTM posee dos características

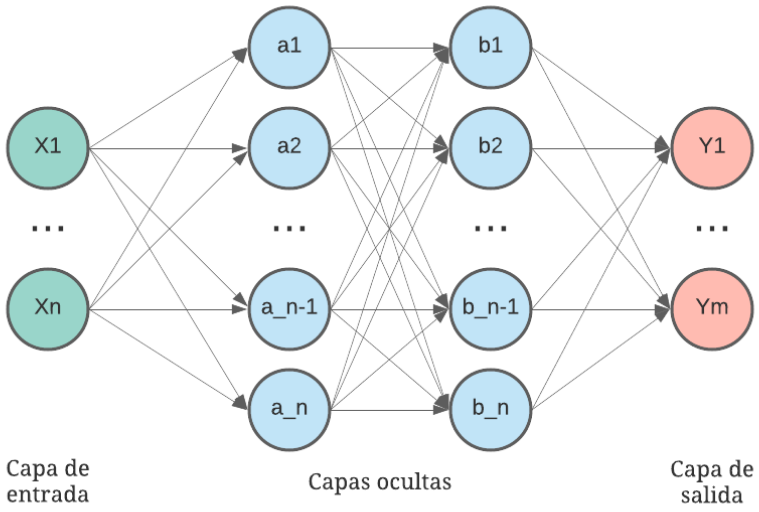


Figura 2.2: Feedforward de múltiples capas.

importantes: 1) un estado explicito que separa conjuntos de variables que almacenan los elementos necesarios para construir las dependencias a largo y corto plazo, y 2) la presencia de compuertas que modulan la cantidad de información que pasa por ellas. En cambio, el modelo GRU, que puede ser considerado como una versión simplificada de LSTM, contiene las compuertas que limitan el flujo de información pero no contienen un estado explícito. Para ambos casos conforme los instantes de tiempo evolucionan, se cuenta con una representación en la que cada capa tiene una dependencia con su predecesora (Figura 2.3).

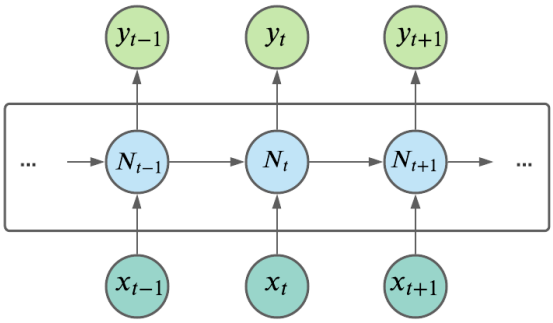


Figura 2.3: Evolución de una RNN simple en el tiempo.

**Redes Neuronales Convolucionales (CNN por sus siglas en inglés).** Aunque usadas principalmente en el análisis de imágenes, en los últimos años han iniciado su incursión a NLP (Hu et al., 2014) y (Albawi et al., 2018). Consisten en redes que sintetizan la información entre capas a través de operadores convolucionales. En su configuración básica contienen varios grupos de capas: capa convolucional, capa de no linealidad, capa de pooling, y la capa conectada, Figura 2.4.

Los modelos descritos por lo general son acuñados bajo la categoría de Aprendizaje Profundo (AP). Estos se encuentran compuestos por múltiples capas de procesamiento (capas ocultas) para

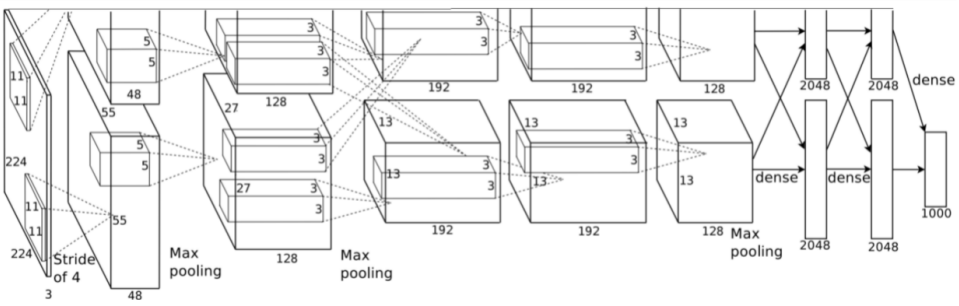


Figura 2.4: Red Neuronal Convolucional por Krizhevsky et al. (2012).

aprender representaciones de los datos con múltiples niveles de abstracción. A través del AP se descubren estructuras complejas en grandes conjuntos de documentos mediante el uso del algoritmo de *backpropagation* que en términos generales conduce el aprendizaje del modelo indicando como debe cambiar sus parámetros de acuerdo a los datos (Lecun et al., 2015).

## 2.8. Redes Neuronales Recurrentes (RNNs) para la Generación de Textos

En algunos problemas el contexto es de suma importancia para predecir una salida; por ejemplo, para predicciones en series de tiempo o el procesamiento de textos. En estos casos, se necesita que los modelos tomen como entrada toda una secuencia antes que entradas aisladas, y además se requiere una retroalimentación entre sus procesos internos de modo que exista una dependencia entre las entradas. Una arquitectura que ha brindado una solución a este tipo de problemas consiste en las RNNs (Bonaccorso et al., 2018).

En su representación más sencilla introducida por (Elman, 1990) una RNN toma una entrada $X_t$ para producir una salida $Y_t$. Integran un factor de recurrencia de modo que en su siguiente iteración la entrada contenga a $Y_t$ junto con la entrada $X_{t+1}$ (Figura 2.5). De este modo, en el instante $t_1$ dentro de la neurona de la capa oculta se tiene una entrada $x_1$ junto con $y_0$.

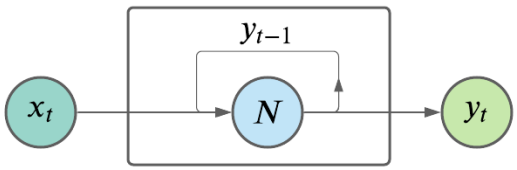


Figura 2.5: Representación de una RNN simple.

Aunque esta arquitectura tiene gran poder al momento de modelar sistemas dinámicos, su entrenamiento se ha demostrado problemático debido a que los gradientes resultado de cada etapa del algoritmo de *backpropagation* o crecen o decrecen, resultando luego de varias etapas en una explosión o desvanecimiento (Lecun et al., 2015). Debido a esta problemática y para acomodarse a otras necesidades

se han extendido otras arquitecturas a partir de la versión simple, entre las principales (Bonaccorso et al., 2018): Bidirectional RNN (BRNN), Deep Bidirectional RNN (DBRNN), Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU).

### 2.8.1. Gated Recurrent Unit (GRU)

Introducidas por Cho et al. (2014), consisten en dos redes neuronales recurrentes. La primera se encarga de codificar una secuencia de símbolos en una representación de longitud fija; mientras que la segunda decodifica la secuencia en otra secuencia de símbolos. Ambas redes son entrenadas de forma conjunta para maximizar la probabilidad condicional de una relación entrada - salida. A esta arquitectura los investigadores introducen una unidad oculta que actúa como las compuertas de la red LSTM, pero con una composición simplificada de modo que logran mejorar la memoria de la red y su facilidad de entrenamiento.

La unidad oculta, incluye en cada neurona de la capa oculta y se conforma por dos compuertas: de actualización $z$ (*update gate*) y de restablecimiento $r$ (*reset gate*); arquitectura presentada en la Figura 2.6. Estas compuertas son las encargadas de decidir qué información debe pasar por cada estado oculto $h$ (*hidden state*) hacia un nuevo estado $\tilde{h}$. De este modo, cada neurona aprenderá a capturar las dependencias sobre diferentes intervalos de tiempo.

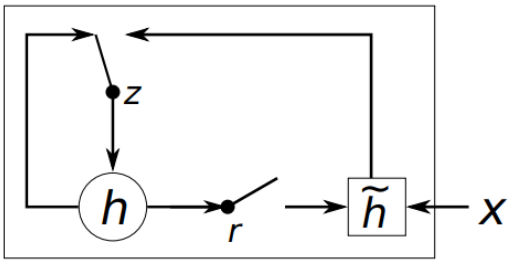


Figura 2.6: Ilustración de la unidad oculta de activación por Cho et al. (2014)

## 2.9. Frameworks y Librerías para Aprendizaje Profundo (AP)

Existen varias tecnologías para trabajar en el campo de AP, Nguyen et al. (2019) mencionan como las más populares a TensorFlow, Keras, Microsoft CNTK, Caffe, Caffe2, Torch, PyTorch, MXNet, Chainer y Teano. A continuación, se realiza una descripción de los tres frameworks/librerías que se consideran para la presente investigación debido a que son de código abierto, la gran cantidad de documentación disponible, soporte a largo plazo de sus módulos, y que su principal lenguaje para interactuar es a través de Python (también se ofrece interfaces para C++, Java, Haskell, R, entre otros).

**TensorFlow.** Framework desarrollado por *Google* (TensorFlow, 2021a) para proporcionar una plataforma que facilite la compilación e implementación de modelos de AA. Ofrece varios niveles de abstracción para adaptarse a diferentes necesidades. Una de sus fortalezas consiste en la *API de estrategia de distribución*, con la que se puede entrenar de forma distribuida a un modelo en diferentes configuraciones de hardware sin necesidad de cambiar la definición del mismo.

**Keras.** Librería construida sobre TensorFlow que simplifica el trabajo a través de un diseño orientado hacia la usabilidad de las personas. Busca reducir el número de acciones requeridas

por los usuarios en los casos más comunes tanto como para el entrenamiento de modelos como integraciones con hardware como clústers de GPUs, pero también permite el desarrollo en bajo nivel (Keras, 2021).

**PyTorch.** Desarrollada por *Facebook* a partir del framework Torch se describe como una librería optimizada de tensores para Aprendizaje Profunto (AP) utilizando GPUs y CPUs. Además de entregar una versión estable que posee mantenimiento a largo plazo dispone de una versión Beta donde se realizan modificaciones de acuerdo a la retroalimentación de la comunidad. Además de permitir un entrenamiento distribuido de modelos, tiene gran facilidad de despliegue en la fase de producción (PyTorch, 2021).

Capítulo 3

# Estado del Arte y Trabajos Relacionados

La sistematización de referentes conceptuales se divide en dos niveles, el primero referido a la construcción y evaluación de corpus cuyos documentos se basan en una o más oraciones; esto debido a que también existen corpus basados en palabras como unidad de análisis. Y, el segundo nivel consiste en los modelos generacionales de textos.

## 3.1. Construcción y Evaluación de Corpus

La importancia de un corpus radica en que permite el entendimiento de la naturaleza del lenguaje (Chafe, 2011), para su construcción los investigadores se han valido de diversas técnicas. Koehn (2005) crearon un corpus paralelo en 11 lenguajes a través de la recolección de las actas del parlamento europeo, las cuales eran traducidas por agentes gubernamentales; este corpus es uno de los más utilizados en el desarrollo de investigaciones de NLP. Sobre este corpus Graën et al. (2014) realizaron una inspección y limpieza, implementando un formato XML para facilitar la selección más sofisticada de los datos. Para el proceso de limpieza compararon a las palabras frente a insumos oficiales del uso del lenguaje como por ejemplo diccionarios. Además, a través de un conjunto de reglas corrigen caracteres incorrectos, signos de puntuación mal implementados y eliminan urls. Entre sus resultados indican en primer lugar que, en alta medida, el corpus contaba con palabras omitidas y con metadatos marcados como parte del texto. En segundo lugar, se encontraron etiquetas HTML no codificadas, variantes de caracteres, palabras con faltas ortográficas y texto parcialmente tokenizado. Por último, en baja medida observaron una codificación UTF-8 inválida, signos especiales mal implementados, comentarios no traducidos y texto marcado como metadatos.

Pak y Paroubek (2010) presentaron una técnica para construir corpus en inglés haciendo hincapié en que su metodología se adapta a otros idiomas. Los investigadores crearon un corpus de 300 mil documentos para el análisis de sentimientos y minería de opiniones mediante la recolección de tuits. El proceso para recolectar y filtrar a los documentos se basó en la identificación de emoticones dentro de los mensajes recuperados de 44 medios de comunicación populares. En su metodología realizaron un análisis lingüístico y estadístico del corpus generado; por un lado analizan la distribución de frecuencias de palabras, y por otro la distribución de etiquetas gramaticales (tags) asignadas mediante un TreeTagger. Habiendo generado el corpus, entrenaron un modelo para clasificar sentimientos de

los documentos en tres niveles: positivo, neutral y negativo. En este estudio si bien se desarrolla un análisis al corpus final, no se realiza una evaluación de la calidad de los documentos que lo conforman.

Cao et al. (2010) y Eisele y Chen (2010) construyeron un corpus paralelo aprovechando la información en varios idiomas que se presenta dentro sitios web. Utilizaron la información disponible en la página web de las Naciones Unidas, que contiene traducciones de alta calidad en seis idiomas. Sin embargo, como los autores, indican esta técnica presenta la dificultad de trabajar con idiomas que contienen estructuras sintácticas muy diferentes como es el caso del inglés y el chino.

Balahur et al. (2013) construyeron un corpus con aproximadamente 8 mil documentos en inglés a partir de tuits. Realizaron una etapa de limpieza al tokenizar las palabras y compararlas con el Tesauro de Roget, el cuál fue elaborado para ayudar a los escritores a elegir palabras apropiadas (Kilgarriff y Yallop, 2001); en caso de no existir coincidencia se buscó similaridad hasta en dos caracteres para poder reemplazar a la palabra. De este modo la palabra perrrfect, se remplazó por perfect. A cada texto lo etiquetaron de forma manual con un sentimiento. Luego, mediante Google Translate, tradujeron al corpus de forma automática a cuatro idiomas, incluido el español. Dividieron a los documentos en subconjuntos de entrenamiento, validación y evaluación; en este último grupo corrigieron las traducciones nuevamente de forma manual, y lo usaron como el Gold Standar. Para cada idioma entrenaron a un modelo clasificador de sentimientos y ponderaron los resultados individuales obteniendo un mejor valor de exactitud. El principal problema de esta metodología consiste en su dificultad de escalar; requiriendo de altos recursos humanos.

Balahur y Turchi (2014) reconocen una carencia en el desarrollo de insumos para lenguajes diferentes al inglés. Proponen una metodología para construir corpus a partir de la traducción de textos por medio de sistemas automáticos: Bing Translator, Google Translate y Moses. De este modo, a partir de un corpus en inglés con 12 mil documentos generaron tres adicionales: en alemán, español y francés. Para evaluar el resultado, analizaron de forma manual un conjunto de documentos traducidos, llegando a la conclusión de que los sistemas traductores para aquella época poseían un nivel de madurez razonable para ser utilizados en ciertos lenguajes. Un factor que complica la implementación de esta metodología consiste en la cuota de uso límite que establecen las empresas propietarias de estos sistemas traductores, creando una complejidad de recursos económicos para llevar a cabo esta tarea a gran escala.

Bowman et al. (2015a) crearon un corpus balanceado de gran escala en inglés denominado *Stanford Natural Language Inference* (SNLI), el cuál posee 570 mil pares de oraciones escritas y etiquetadas por alrededor de 2500 trabajadores contrados a través de AMT. Cada par de oraciones contiene implicación, neutralidad o contradicción semántica. El objetivo de este corpus buscó evaluar modelos clasificadores de textos. A través de este conjunto de datos, lograron alimentar por primera vez para esta tarea a modelos basados en redes neuronales y superaron los resultados de los clasificadores existentes (Bowman et al., 2015a). Dentro del procesamiento de datos realizaron un análisis de la longitud de las oraciones, y dividieron al conjunto total en segmentos de entrenamiento, validación y evaluación. Utilizando nuevamente la plataforma AMT evaluaron el 10 % de los documentos a través de una prueba que buscaba determinar si el par de oraciones creado inicialmente era correcto.

Minard et al. (2016) presentaron un corpus basado en noticias del portal Wikinews y lo denominaron MEANTIME. Su propósito consiste en apoyar la construcción de un sistema de reconstrucción de líneas de historias a través de varias noticias de modo que se pueda presentar un panorama general de lo ocurrido. Cuenta con 120 documentos disponibles en 4 idiomas. Realizaron anotaciones manuales en diferentes niveles como entidades, eventos y tiempo de expresiones. Y, posteriormente, crearon relaciones entre las anotaciones del nivel anterior. Dentro de su formato de presentación combinan XML para el texto y las anotaciones, y CSV para los metadatos.

En otros casos, los esfuerzos para construir un corpus provienen de un esfuerzo conjunto de varias personas y organizaciones. Sosoni et al. (2018) a través de crowdsourcing tradujeron a 11 idiomas 87 mil segmentos de textos (oraciones); el dominio de los documentos corresponde al ámbito académico, provenientes de plataformas Massive Open Online Courses (MOOCs). La traducción se realizó de forma manual; en este proceso participaron más de 2 mil voluntarios a lo largo de cuatro meses. La limpieza a los documentos iniciales consistió en la eliminación de los metadatos y la conversión de los caracteres a la codificación UTF-8. Para controlar la calidad del corpus final, las traducciones fueron evaluadas constantemente a través de muestreos, y cuando se determinaba que la calidad de las traducciones de un voluntario era baja, se lo reemplazaba por otra persona. Un inconveniente con el resultado final consiste en que se encuentra sujeto a varios tipos de restricciones; al haber participado organizaciones de varios países, las licencias y restricciones de uso van de acuerdo a la fuente que proporcionó los datos.

Williams et al. (2018) identificaron limitaciones en el corpus SNLI refiriendo que no contempla varios temas, estilos y grados de formalidad. Ante esto crearon un corpus que integre estos aspectos de manera que pueda ser utilizado como benchmarking para modelos de Machine Learning, proporcionando además este insumo para facilitar las tareas de aprendizaje por transferencia de conocimiento entre dominios. Al resultado lo denominaron Multi-Genre NLI Corpus (MultiNLI), el cual contiene 433 mil documentos basados en oraciones. Para su construcción y evaluación, los investigadores utilizaron la misma metodología empleada para elaborar el corpus SNLI. Al final, realizaron una comparación entre tres modelos entrenados a partir de este corpus y del corpus SNLI. Los modelos entrenados en el corpus MultiNLI obtuvieron métricas inferiores en alrededor del 15 %, demostrando que su conjunto de documentos contiene una mayor complejidad.

A partir del corpus MultiNLI, Conneau et al. (2018) tomaron un conjunto de 10 mil documentos y los tradujeron de manera manual a 15 idiomas; al corpus resultante lo llamaron XNLI. Para realizar la traducción se utilizó la plataforma One Hour Translation con un costo aproximado de 21 centavos de dólar por palabra traducida para la mayoría de lenguajes (One Hour Translation, 2021). Los investigadores argumentan que traducir documentos existentes antes que generar textos a través de un grupo de trabajadores tiene la ventaja de asegurar que la distribución de los datos sea similar a través de los múltiples lenguajes. Para evaluar la calidad del resultado obtenido, profesionales traductores se encargaron de volver a anotar 100 ejemplos comparando su concordancia con los resultados originales.

Lewis et al. (2019) presentan un corpus altamente paralelizado en siete idiomas llamado MLQA; contiene 42 mil documentos. Busca servir como base para benchmarks de sistemas multilingües de preguntas y respuestas (QA). Para su construcción recolectaron documentos en inglés del sitio web Wikipedia; buscaron que el dominio sea variado y que este tenga similitud a los recursos ya disponibles para tareas de QA. Se apoyaron de la plataforma AMT para generar preguntas sobre los documentos obtenidos juntos con las respuestas (extractos del documento original). A partir de este resultado, utilizaron técnicas de minería de datos en las oraciones a modo de detectar duplicaciones y así evitar el incremento innecesario en el costo que además genera un corpus con documentos no naturales. Posteriormente, tradujeron a través de profesionales los documentos a los idiomas objetivos.

Construido y provisto por el medio de comunicación Reuters, se encuentra el corpus RCV2 (Reuters, 2019). Cuenta con más de 487 mil documentos no paralelizados en 13 idiomas y consisten de historias escritas por reporteros locales de *Reuters News*. A diferencia de los corpus abiertos mencionados con anterioridad, el acceso a esta información es restringido; se debe realizar un acuerdo entre la organización solicitante con *Reuters*. Estos datos los proporcionan únicamente para investigaciones de NLP, e indican que una vez concluido el acuerdo la parte solicitante tiene la obligación de eliminar la

información provista. De acuerdo a Keung et al. (2020), este corpus, a más de ser de difícil acceso, es altamente desbalanceado y no provee conjuntos de entrenamiento, validación y evaluación.

Ni et al. (2020) buscaron generar justificaciones automáticas frente a una opinión como entrada. Para esto construyeron un corpus con más de 1 millón de documentos al combinar un sub grupo del set de datos Yelp con el set de datos Amazon Clothing. El primer conjunto es entregado por la empresa estadounidense Yelp, en su versión original contiene más de 8 millones de documentos; mientras que, el segundo conjunto consiste en una categoría del corpus completo entregado por la empresa Amazon.

Varias investigaciones se han llevado a cabo para reducir la brecha entre los insumos en inglés y otros idiomas. Keung et al. (2020) crearon un corpus multilingüe de gran escala a partir corpus provistos por la empresa privada Amazon. Los investigadores recolectaron documentos (con fecha de publicación entre 2015 a 2019), y a través de algoritmos de detección de lenguaje generaron sub conjuntos para 6 idiomas, incluido el español; cada uno de estos conjuntos contiene 210 mil documentos y se encuentra con particiones para entrenamiento, validación y evaluación (200 mil, 5 mil y 5 mil respectivamente). Posteriormente, aplicaron técnicas de muestreo, filtrado y procesamiento de textos para eliminar documentos considerados como outliers. Una de las técnicas utilizadas para la limpieza de los textos consistió en convertir los textos a la codificación UTF-8 y eliminar etiquetas HTML. A través de estos resultados entrenaron un modelo para clasificar al texto de los documentos de acuerdo al número de estrellas. Si bien los resultados son de alta relevancia, no se presentan métricas para evaluar al corpus generados, sino más bien para evaluar los resultados de la clasificación del modelo.

En algunas situaciones la construcción de los corpus se simplifica debido a que existe un trabajo previo. Cattoni et al. (2020) elaboraron un corpus basado en los subtítulos de las charlas de la organización TED; a cada documento lo relacionaron con sus equivalente en cada uno de los idiomas a los que la organización traduce de forma manual. El resultado consiste en el corpus denominado MuST-C que contiene 270 mil oraciones traducidas a 14 idiomas. A pesar de que se podría inferir una alta calidad del corpus debido a que todas las anotaciones fueron realizadas por humanos, los investigadores analizan de forma manual el 20 % de los registros.

Scialom et al. (2020) introducen al corpus MLSUM cuyo objetivo es proveer un insumo de gran escala para tareas de resúmenes automáticos. Consta de 1.5 millones de pares de documentos: artículo/resumen en 5 idiomas diferentes, incluido el español. Para su construcción los investigadores recolectaron a través de web scraping la información de portales de noticias en un intervalo de 9 años; aprovechando que en la mayoría de artículos se cuenta con una sección de resumen que acompaña al texto principal. La limpieza al conjunto de datos consistió en conservar documentos que superen un número predefinido de palabras. La división para entrenamiento, validación y evaluación la realizan de acuerdo a los años: 2010 a 2018 para entrenamiento y 2019 para validación y evaluación. Sin embargo, esta metodología entra en conflicto con lo indicado por Liu y Han (2012) quienes mencionan que el uso del lenguaje cambia y en la fase de diseño experimental se debe conservar una distribución uniforme en cuanto a temporalidad.

También existen corpus multilingües etiquetados como masivos. Los dos casos más representativos de esta categoría corresponde a XGLUE (Liang et al., 2020) y XTREME (Hu et al., 2017). En ambos casos la capacidad de los corpus es tan extensa que ya no se hace referencia a la cantidad de documentos, sino al tamaño computacional: 15 GB para XTREME (su archivo comprimido) y 2 TB para XGLUE. En esencia estos corpus consisten en una integración de varios corpus de gran escala como XNLI, MLQA, entre otros, aunque también cuentan con material obtenido a través de *web scraping*. La función de estos conjuntos de documentos consiste en servir para el benchmarking de múltiples tareas como generación de textos, traducción automática, comprensión de textos, y más.

Existen corpus construidos y provistos por empresas privadas; dos representativos corresponden a los provistos por Amazon y Yelp, los cuales fueron construidos a partir de opiniones de los usuarios de las plataformas. Amazon, el retailer más grande del mundo (Forbes, 2019), provee el corpus Amazon Customer Reviews (Amazon, 2020) que contiene más de 130 millones de comentarios en varios idiomas, escritos desde el año 1995 hasta 2015. Incluye a las plataformas de ventas de cinco países, aunque no cuenta con la especificación del lenguaje en que se encuentra escrito el documento. Por el lado de la empresa Yelp, cuya plataforma permite a los usuarios proporcionar recomendaciones acerca de restaurantes, compras, comida, entretenimiento, entre otros (Yelp, 2021), se entrega el insumo denominado Yelp Open Dataset (Yelp, 2019), el cual dispone de más de 8 millones de comentarios, de igual manera que en anterior caso se incluyen comentarios en varios idiomas aunque no se dispone de un identificador del lenguaje. Ambos corpus se entregan bajo los términos de uso de cada empresa, refiriendo un acceso abierto de uso no comercial, para propósitos académicos.

## 3.2. Generación automática de textos

Existen grandes dificultades dentro de la tarea de generación automática de textos. Li et al. (2018b) indican que la mayoría de modelos no puede rendir de forma esperada cuando el tamaño de los cuerpos de datos es reducido, además hacen referencia a la existencia de un hueco entre la importancia de tener un gran dataset para entrenamiento y la dificultad para obtenerlo. Por otro lado, Yang et al. (2019) observan que al crear oraciones sintéticas por decodificar muestras aleatorias de un espacio latente, la mayoría de regiones de las que se samplea no necesariamente se traducen a oraciones realistas.

Varias investigaciones se han desarrollado para hacer frente a esta tarea. Por ejemplo, Bowman et al. (2015b) generaron un conjunto de oraciones en base a interpolar dos oraciones entregadas como entrada integrando Variational Auto Encoders (VAE) con RNN-LSTM. Estas oraciones producidas lograron conservar el estilo, el dominio y características sintácticas de alto nivel. Dentro de sus experimentos logran completar oraciones en las que existían palabras faltantes.

Hu et al. (2017) logran generar texto controlando varios atributos: longitud de los documentos, sentimiento y tiempo de las oraciones. Combinan VAEs con Discriminadores Holísticos de Atributos (que podrían ser entendidos como GANs) para mantener la estructura semántica objetivo, y utilizan el algoritmo wake-sleep de modo que se obtenga un espacio latente desenredado, de esta forma los ejemplos evitan convertirse en palabras aleatorias sin sentido.

Li et al. (2018a) proponen un modelo para generar oraciones dentro de una categoría predefinida combinando RNNs, Generative Adversarial Networks (GANs) y Reinforcement Learning (RL). De este modo, además de controlar el dominio de generación, pueden incrementar el tamaño original del corpus; logrando mejores niveles de generalización durante el entrenamiento supervisado.

Logeswaran et al. (2018) desarrollan un modelo capaz de generar oraciones compatibles a un conjunto de atributos condicionales: sentimientos, complejidad del lenguaje, tiempo, voz y humor; preservan el contenido original a través de Auto Encoders y Back-Translation Losses controlados por un discriminador adversarial. Para la fase de evaluación incorporan métricas para determinar de forma objetiva el grado en que el modelo conserva la compatibilidad de los atributos.

Brown et al. (2020b) mencionan sobre una tendencia en modelos del lenguaje basados en *transformadores*. Resaltan modelos como RNSS18 con 100 millones de parámetos, DCLT18 con 1.5 billones de parámetos y Tur20 con 17 billones de parámetros. Dentro de su investigación presentan a GPT-3, un modelo generacional con 100 billones de parámetros, el cual es capaz de aprendizaje "zero shot.[en] el cual a través de indicaciones basadas en lenguaje natural el modelo es capaz de entender la orden y

producir un resultado.

Prabhumoye et al. (2020) indican que un problema con la generación de textos consiste en que este texto es altamente aleatorio e incontrolable. Para hacer frente a esta problemática mencionan que se deben aprender representaciones desenredadas del lenguaje en donde cada parte del modelo aprenda una característica del espacio latente; de esta forma se pueden entrenar modelos capaces de generar documentos interpretables. Dentro del modelo incorporan un generador y discriminador que se encuentran en constante retroalimentación y muestran que con poca supervisión se puede aprender representaciones estructuradas

CAPÍTULO 4

# Diseño e Implementación

El presente capítulo presenta el diseño metodológico para la construcción y evaluación del corpus, así como los criterios para su implementación. En primer lugar, se caracteriza el corpus que se busca obtener. En segundo lugar, se seleccionan las fuentes de los documentos sobre las que se realizaran las recolecciones de datos. En tercer lugar, se designan las métricas para realizar la evaluación automática, y se define el diseño experimental para la evaluación humana. En cuarto lugar, se realiza el diseño de los sistemas darán soporte a la tarea de construcción a través de la recolección, limpieza, traducción, evaluación y generación de resultados de línea base. Por último, se detallan los criterios sobre los que se implementará la metodología descrita; en la Figura 4.1 se presenta el esquema general del modelo desarrollado para la construcción y evaluación del corpus objetivo.

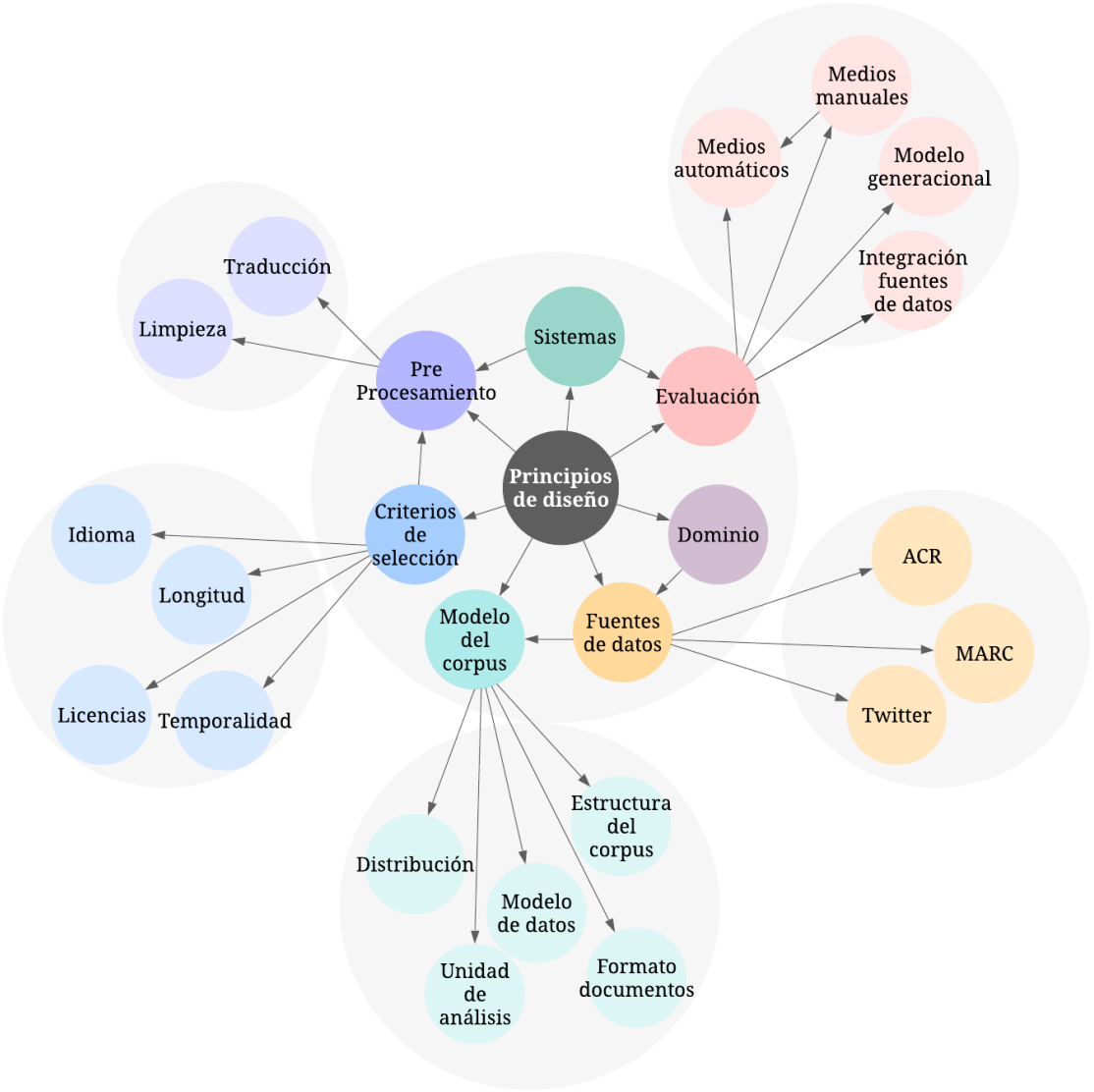


Figura 4.1: Modelo para la construcción y evaluación del corpus en español.

## 4.1. Caracterización del Corpus Objetivo

Se presentan las características principales que deberá poseer el corpus a construir para que este sirva a modelos de AP que tengan la capacidad de generar comentarios dentro del dominio establecido.

### 4.1.1. Definición del Dominio del Corpus

Como se indicó en el Capítulo 1, la presente investigación tiene el respaldo del proyecto **SUMA**, buscando brindar apoyo a sus objetivos. Se ha definido al dominio del corpus como de comentarios de productos textiles. Das (2009) indica que son considerados como productos textiles aquellos que cumplen una de las siguientes propiedades.

- Productos definidos como crudos, semi trabajados, semi manufacturados, manufacturados, semi hechos o hechos de productos que son exclusivamente compuestos por fibras textiles sin importar sus combinaciones o procesos de elaboración.
- Productos que contienen al menos 80 % de su peso compuesto por fibras textiles.
- Cobertores de piso, muebles, sombrillas, cobertores solares, colchones, zapatos y guantes que contengan al menos el 80 % de su peso compuesto por fibras textiles.

Albán et al. (2020) indican que en Ecuador destacan dentro de la industria textil los productos de vestir, lencería de hogar y lencería para negocios. Dentro de este contexto, se buscará recolectar información sobre los siguientes productos:

- Prendas de vestir: abrigos, calcetines, camisetas, camisas, chompas, casacas, chaquetas, faldas, pantalones, ropa de dormir, vestidos ternos, sacos, medias, zapatos, ropa interior, buzos, gorras, polos, pijamas, sudaderas y guantes.
- Lencería para el hogar y negocios: cobertores para exterior, cobertores de cama, cobertores para muebles, sábanas, cortinas, manteles, toallas, alfombras, tapetes, mandiles, uniformes, edredones y colchas.

A partir de estas palabras se han generado reglas de búsqueda para obtener documentos de las fuentes. Estas reglas se basan en la siguiente estructura:

$$Sustantivos, Verbos, Adjetivos$$

Dentro de cada grupo de palabras existe una conjunción entre ellas, mientras que una disyunción entre cada categoría. De este modo, al expandir la estructura general de la regla se obtiene:

$$\begin{aligned}&(Sustantivo\ 1\ \ OR\ \ Sustantivo\ 2\ \ OR\ \ \ldots\ \ OR\ \ Sustantivo\ P)\ \ AND\\&(Verbo\ 1\ OR\ \ Verbo\ 2\ \ OR\ \ \ldots\ \ OR\ \ Verbo\ Q)\ \ AND\\&(Adjetivo\ 1\ \ OR\ Adjetivo\ 2\ OR\ \ldots\ \ OR\ Adjetivo\ R)\end{aligned}$$

Para cada grupo de palabras se generó una combinación tomando en consideración palabras singulares, plurales, sinónimos y posibles errores ortográficos referentes a la omisión de tildes. En el Anexo C se presentan las reglas utilizadas de acuerdo a cada fuente de datos.

### 4.1.2. Principios de Diseño

Un corpus de utilidad no consiste en una serie de documentos recolectados de forma aleatoria. Requiere que sus documentos se encuentren relacionados en algún aspecto de modo que se puedan

considerar como parte de una misma distribución. De esta distribución un modelo generacional aprenderá patrones que podrá replicar en nuevos contextos. A continuación, se presentan los principios de diseño que regirán la construcción del corpus.

**Dominio específico de los documentos.** Se busca generar un insumo que permita el entrenamiento de un modelo de enfoque específico.
**Características de los documentos.** Los documentos originalmente deben ser escritos por personas. En caso de no poseer dentro de su contenido palabras relacionadas a productos textiles, deberán provenir de fuentes de datos que provean documentos del dominio establecido.
**Cantidad de documentos.** Existe cierta ambigüedad al referirse al tamaño que un corpus debe tener. O'Keeffe y McCarthy (2010) indican dos factores para establecer el tamaño del corpus: representatividad (que la cantidad de documentos logre captar el problema) y practicidad (que el tiempo de recolección sea acorde a los recursos de la investigación). Bajo este contexto se establece como condición para finalizar la recolección de documentos: 1) Obtener un mínimo de 200 mil documentos, asemejando el diseño de los corpus MARC, MuST-C, y MLSUM, presentados por (Keung et al., 2020), (Cattoni et al., 2020), y (Scialom et al., 2020) respectivamente.
**Diversidad.** Los documentos dentro del corpus si bien son de dominio específico deberán representar diferentes semánticas y poseer diferentes estructuras sintácticas. Esto además de mejorar el entrenamiento del modelo generacional, evita redundancia durante las tareas de limpieza y traducción.
**Anonimidad.** Se preservará la identidad de los usuarios creadores de los documentos. No se almacenarán atributos que puedan comprometer su identidad.
**Accesibilidad.** El corpus final debería ser de acceso abierto. La única licencia adicional a la de las fuentes de datos que se debería considerar es CC BY, bajo la motivación de proporcionar un recurso abierto para otros investigadores.

### 4.1.3. Criterios de Selección de Documentos

Se establecen los siguientes criterios al momento de recolectar los textos de tal manera que se delimite el alcance, se eliminen documentos que podrían causar ruido y se elijan documentos con licencias que permitan extender funcionalidades.

**Idioma.** Si bien se propone una metodología que se puede adaptar a varios idiomas, se plantea dentro de esta investigación trabajar con documentos escritos en español e inglés.
**Licencia.** Los documentos provendrán de fuentes cuyos conjuntos de datos se encuentren bajo las siguientes licencias: CC0, CC BY, CC BY-SA, CC BY-NC, CC BY-NC-SA o MIT. Se podrán obtener documentos de fuentes personalizadas siempre que estas tengan un acceso abierto y se asemejen a las descritas anteriormente. Se evita el uso de licencias que no permitan o limiten las modificaciones a los datos originales.
**Temporalidad.** Al igual que Liu y Han (2012) se busca conservar expresiones del momento, por lo que se adoptará un intervalo de 10 años. Los documentos deberán haber sido publicados entre el año 2011 (tomando como referencia el estudio de Keung et al. (2020)) a la fecha actual del desarrollo del trabajo de investigación.
**Longitud mínima.** Se conservan los documentos que contengan al menos cinco palabras de longitud; de este modo se eliminan comentarios con poca expresividad como por ejemplo *todo bien*, *ok* y *no lo recomiendo*.

### 4.1.4. Modelo del Corpus

A continuación, se introducen las características del modelo del corpus a construir. Se define el diseño estructural del insumo así como el formato que dispondrán los documentos.

**Unidad de análisis.** El texto dentro de cada documento se conformará de una o varias oraciones preservando la puntuación; contendrá como mínimo 5 palabras de longitud y como máximo 120, textos con mayor longitud serán truncados en este límite; si bien se podría pensar que este trucamiento ocasionaría una pérdida del significado de la oración, se plantea la suposición de que la idea central del comentario se encontrará al inicio del mismo. El valor máximo de palabras se establece debido a la tarea de traducción, la cual se encuentra limitada por cuotas de uso de las diferentes plataformas traductoras.

**Modelo de datos.** Cada documento dentro del corpus contendrá varios tipos de atributos. Los que se marcan como obligatorios consisten en elementos indispensables para validar a un documento. Por otro lado, los no obligatorios no son requeridos para anexar a un documento al corpus ya que dependen principalmente de las características de las fuentes de datos; se los incluyen debido a que conservan información que puede ser aprovechada en futuras líneas de investigación. En la Figura 4.2 se presenta el modelo desarrollado para el presente estudio; se detallan cada uno de sus atributos.

- **Id:** Número de identificación del documento dentro del corpus. Consiste en una cadena de texto con valores numéricos creada a partir de la fecha de recolección en nanosegundos; campo obligatorio.
- **Fuente:** Cadena de texto con el nombre de la fuente de la que proviene el documento; campo obligatorio.
- **Fecha de publicación:** Fecha que indica la fuente en que el documento fue creado, campo no obligatorio.
- **Fecha de obtención** Fecha en que se recolectó el documento en formado *día mes año*; campo obligatorio.
- **Anotaciones:** Conjunto de anotaciones provistas en la fuente original; campo no obligatorio. El objetivo de este atributo es conservar información previa que pueda extenderse a futuras investigaciones. Para cada anotación se genera un nuevo atributo, de este modo se tendrá el siguiente formato: *anotaciones.anotacion_1*, *anotaciones.anotacion_2*, ..., *anotaciones.anotacion_n*.
- **Lenguaje original:** Especificación sobre el lenguaje original del documento: “en” para inglés y “es” para español; campo obligatorio.
- **Traductor:** En caso de haber sido traducido el documento por medios automáticos se coloca el nombre del sistema traductor; campo no obligatorio.
- **Truncado:** Indicador sobre si el texto fue o no truncado al llegar a 120 palabras de longitud; campo obligatorio.
- **Título:** Cadena de texto que refiere al título de un comentario en caso de que la fuente lo provea; campo no obligatorio.
- **Texto:** Cadena de texto que contiene el comentario referente a un producto textil; campo obligatorio.

**Formato de los documentos.** Se ha seleccionado a *JSON Lines* como el formato de los documentos debido a su facilidad de lectura tanto nivel humano como máquina sin de un

| Documento | |
|---|---|
| id | str |
| fuente | str |
| fechaPublicacion | Date |
| fechaObtencion | Date |
| anotaciones | str |
| lenguajeOriginal | str |
| traductor | str |
| truncado | bool |
| titulo | str |
| texto | str |

Figura 4.2: Modelo de datos para los documentos del corpus.

procesamiento previo para su lectura. La Figura 4.3 presenta una muestra de un documento ejemplo bajo este formato.


```
{
"id": "093812749823908741",
"fuente": "Twitter",
"fechaPublicacion": "7 06 2016",
"fechaObtencion": "9 5 2021",
"lenguajeOriginal": "en",
"traductor": "googletrans",
"truncado": false,
"titulo": null,
"texto": "Me gustaron las costuras que...",
"anotaciones.possible_sensitive": null,
"anotaciones.stars": null,
"anotaciones.overall": 4,
}
```


Figura 4.3: Muestra de un documento en su formato.

**Estructura del corpus.** La estructura del corpus dentro del sistema de archivos consistirá en una carpeta que contendrá tres subcarpetas: entrenamiento, validación y evaluación. Dentro de cada una se encontrarán los archivos en formato *.json*. La convención para nombrar a los archivos consiste en la palabra *corpus*, seguido de la *fase de destino*. En la Figura 4.4 se presenta un ejemplo de la convención utilizada.

**Medio de distribución del corpus.** Se elige a la plataforma *GitHub* como medio para distribuir el corpus final, debido a su amplia acogida por investigadores para almacenar estos insumos. Si bien posee una restricción de 100MB por archivo; el resultado final en su versión comprimida tiene un tamaño menor a este límite.

### 4.1.5. Selección de Fuentes de Documentos y Diseño de Integración de los Modelos de Datos

Cada año incrementa el número de corpus disponible para los investigadores; por ende, O'Keeffe y McCarthy (2010) recomiendan que antes de construir un corpus se debe estar seguro que dicho insumo

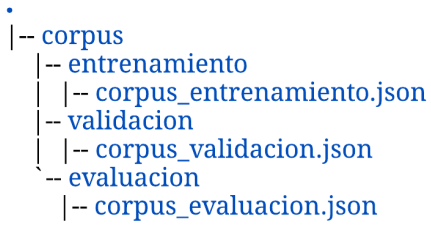


Figura 4.4: Convención para estructurar al corpus.

no existe, y buscar aprovechar los insumos existentes para su construcción. Bajo esta premisa, se propone un enfoque híbrido de construcción a través de la recolección de documentos dentro de redes sociales, junto con la integración de subconjuntos de documentos de corpus ya existentes y relevantes al objetivo del estudio. Se han elegido tres fuentes para obtener los documentos y de estas se han seleccionado los atributos relevantes al corpus a construir. Para información sobre todos los atributos proporcionados por cada fuente refiérase al Anexo A.

**API de Twitter.** Durante el primer cuatrimestre del año 2020 se identificaron 330 millones de cuentas en *Twitter*, de las cuales 186 millones son consideradas como activas (Statista, 2021); no se disponen de datos específicos para Ecuador. Dentro de su sitio web, esta red social detalla dos tipos de cuentas: públicas y protegidas. Las cuentas públicas (activadas por defecto) son visibles para cualquier persona, mientras que para las cuentas protegidas se requiere del permiso del propietario para poder acceder a su información. De acuerdo a una encuesta realizada en Estados Unidos, se estima que alrededor del 13 % de los usuarios del país en mención mantienen su cuenta en modo protegido (Remy, 2019). Esto evidencia que a través de *Twitter* se puede acceder a la información de la mayoría de sus cuentas. En relación a la cantidad de información generada se estima que en promedio son enviados 200 billones de tuits por año (Sayce, 2020).
Debido a la apertura que existe por parte de *Twitter* a extraer datos, a la gran cantidad de usuarios que existen y a la gran cantidad de información que se genera dentro de su ambiente se la ha elegido como red social destino para el presente estudio. Además, el límite máximo de longitud de los mensajes, 280 caracteres, permite obtener una idea central dentro de un texto corto.
Se recolectarán comentarios a través de una coincidencia de palabras claves. La unidad de carga útil (*payload* en inglés) dentro de la respuesta a un requerimiento a la API de *Twitter* consiste en un documento JSON con varias decenas de atributos (varía de acuerdo al número de veces que ha sido compartido el mensaje). Los atributos que se tomarán en consideración para el modelo propuesto son los siguientes:

- **created_at:** Fecha de creación.
- **full_text:** Texto del mensaje publicado.
- **lang:** Abreviatura para el lenguaje del documento.
- **possible_sensitive:** Referencia sobre si el contenido puede afectar a las personas.

**The Multilingual Amazon Reviews Corpus (MARC).** El corpus MARC tiene la ventaja de contener 200 mil comentarios en español repartidos en 31 categorías, que han pasado por una fase de limpieza. No obstante, tiene la limitante de no disponer de los nombres de los productos

ya que estos se encuentran anonimizados, y además, el número de documentos dentro de las categorías de las que se pueden obtener productos textiles es reducido, en comparación a otras categorías como por ejemplo la categoría de libros.
A partir del corpus general se trabajará en dos niveles: 1) se obtendrán todos los registros dentro de las categorías *Apparel* y *Shoes*, y 2) se realizará un filtrado a partir de palabras claves dentro de las categorías *Home* y *Furniture*, Los atributos que se utilizarán para adaptarlos al modelo de datos propuestos son los siguientes:

- **review_title:** Título que el usuario proporciona junto con su comentario.
- **review_body:** Texto del mensaje publicado.
- **stars:** Valoración que el usuario coloca junto con su comentario.
- **language:** Lenguaje en que el texto del comentario fue escrito.

**Amazon Customer Reviews (ACR).** A diferencia del corpus MARC, el corpus ACR no contiene los nombres de los productos anonimizados, pero los documentos se encuentran en inglés por lo que los registros recuperados de esta fuente se someterán a una posterior fase de traducción. Junto con el corpus se distribuye un documento con los meta datos de cada comentario y de aquí se puede obtener información para identificar el tipo de producto.
Se trabaja con el conjunto de datos denominado *K-cores* en el cual los registros incluidos son los de los usuarios que forman parte de un grafo denso implementado por la empresa *Amazon*. La característica principal de este conjunto de datos consiste en que los usuarios han generado un mínimo número de comentarios en su historial de compras.
Las categorías de análisis para el corpus ACR consisten en *Clothing, Shoes and Jewelry*, y *Home and Kitchen*; los atributos que se conservarán para el modelo propuesto son:

- **reviewText:** Texto del mensaje publicado.
- **overall:** Valoración que el usuario coloca junto con su comentario.
- **summary:** Resumen del comentario, equivalente al título que se dispone en el corpus MARC.
- **reviewTime:** Fecha de publicación del comentario.

Habiendo definido las fuentes de datos a utilizar, se presenta en la Tabla 4.1 la sistematización sobre como los atributos de cada fuente se adaptarán al modelo de datos propuesto, su nombre, y su tipo.

Tabla 4.1: Integración de los atributos de las fuentes al corpus objetivo.

| | **Atributos equivalentes en las fuente** | | |
|---|---|---|---|
| **Atributo del corpus** | **ACR** | **MARC** | **Twitter** |
| lenguajeOriginal | - | language: str | lang: str |
| fechaPublicacion | reviewTime: Date | - | created_at: Date |
| titulo | summary: str | review_title: str | - |
| texto | reviewText: str | review_body: str | full_text: str |
| anotaciones | overall: float | stars: str | possibly_sensitive: bool |

## 4.2. Selección del Modelo Generacional y la Tecnología de Desarrollo

Se implementará un modelo generacional basado en Redes Neuronales Recurrentes (RNNs) entrenado a partir del corpus elaborado que brinde resultados de línea base con respecto a la generación de oraciones automáticas (los detalles de las arquitecturas implementadas se presentan en el siguiente capítulo). De este modo se establecen resultados sobre los cuales se pueden comparar futuros modelos a implementar sobre el mismo corpus. Se seleccionan a modelos basados en esta arquitectura ya que tienen la ventaja sobre modelos probabilísticos de ser más eficientes con el almacenamiento computacional requerido para su entrenamiento (Bensouda et al., 2019).

Se elige la arquitectura GRU sobre LSTM debido a que proporcionan resultados equivalentes sin la complejidad computacional requerida en la red LSTM (Bonaccorso et al., 2018) y (Chung et al., 2014).

La implementación del modelo seleccionado se realizará a través del *framework TensorFlow* junto con la librería *Keras*. Esto debido a ser de código abierto, proporcionar interfaces en Python, tener una gran comunidad que brinda soporte y disponer de amplia documentación.

## 4.3. Pre Procesamiento de Documentos

El pre procesamiento de los documentos consiste en el conjunto de tareas que permiten manipularlos para mejorar la calidad de los datos. De acuerdo a Kotsiantis et al. (2006), esta etapa puede tener un impacto significativo dentro de modelos de aprendizaje supervisado, mejorando la capacidad de generalización. En este apartado se presentan los procesos de limpieza que se aplicaron a cada uno de los documentos.

### 4.3.1. Limpieza

La limpieza de los documentos consiste en la ejecución de un conjunto de procesos para mejorar la calidad de los textos. A través de esto se busca eliminar ruido en los datos, redundancias y documentos que no cumplen los principios de diseño establecidos para el corpus; de cada limpieza se llevará un registro para caracterizarlo en los resultados. Se ejecuta una primera etapa de limpieza, denominada *pre limpieza*, sobre el documento inmediatamente después de su recolección, y antes de ser agregado al corpus en construcción. Esta limpieza consiste en la ejecución de las siguientes tareas sobre el texto:

**Control de longitud mínima.** Se verifica que el documento posea al menos 5 palabras de longitud, caso contrario se descarta.

**Eliminación de Uniform Resource Locators (URLs).** A través de expresiones regulares se eliminan las URLs que se encuentren dentro de los mensajes.

**Eliminación de jerga propia de la plataforma** En el caso de *Twitter* se reemplazan las menciones a otras cuentas por el token *_usr* . Adicionalmente, se sustituyen las palabras etiquetadas como *hashtags* por el token único *_hsh* .

**Limpieza de palabras repetidas.** Se conserva solo una palabra en caso de existir secuencias de palabras contiguas repetidas.

**Limpieza de caracteres especiales repetidos.** Se conserva solo un carácter especial en caso de existir secuencias contiguas de caracteres especiales repetidos.

**Truncamiento del texto.** Si el mensaje posee más de 120 palabras se trunca la longitud en este límite.

**Conversión a minúsculas.** Se convierten a los caracteres en mayúsculas a minúsculas.
**Reemplazo de caracteres numéricos.** A cada caracter o secuencia de caracteres numéricos se reemplazará por el token *_num* .
**Eliminación de emoticones.** Se eliminan los emoticones a través de su caracter *unicode*.
**Reglas de refinamiento.** Se genera un conjunto de reglas para aumentar la calidad del documento. Estas reglas buscan solucionar errores como por ejemplo vocales escritas de la forma *à* o puntuaciones como $palabra_1.palabra_2$.
**Codificación UTF-8.** Se codifica al texto según la codificación UTF-8 para facilitar su manejo entre varias plataformas.

Una vez concluidas las tareas anteriores se procede a almacenar al documento dentro del corpus y se continua con la recolección de documentos. Al finalizar la etapa de recolección se realizará una segunda etapa de limpieza, denominada como *post limpieza*, guardando una copia del corpus original para un control de versiones. Esta limpieza consiste en la ejecución de las siguientes actividades:

**Eliminación de documentos repetidos.** Se eliminan todos los registros con comentarios repetidos.
**Auto corrección de palabras no comunes.** Se buscará reemplazar las palabras con una frecuencia menor de 20 dentro de un diccionario de palabras construido a partir del mismo corpus; este valor se toma como referencia de la investigación realizada por Keung et al. (2020) quienes en vez de buscar reemplazar las palabras, eliminan al comentario por completo. Un modelo probabilístico buscará reemplazar estas palabras catalogadas como infrecuentes por otras que se encuentren a una distancia de edición. De este modo palabras como *corecto* o *correctoo* se podrán reemplazar por *correcto*, no así palabras como *correctooo*.
**Limpieza de documentos con palabras no comunes.** Eliminación de los comentarios que posean al menos una palabra con una frecuencia inferior a 20 dentro del diccionario habiendo pasado por la fase de *Auto corrección de palabras no comunes*.

### 4.3.2. Traducción

La fase de traducción se realiza posteriormente a finalizar la recolección de documentos, antes de la fase de post limpieza. Dentro del modelo de datos propuesto se cuenta con el atributo para indicar el lenguaje original del documento, a partir de este atributo se seleccionarán los textos en inglés y se procedió a su traducción a través de *Cloud Translation* y *Translator Text*. Se implementaron ambas APIs de acuerdo a la cuota disponible. No se trabajan de forma directa con estas APIs debido a que requieren el ingreso de métodos de pago, como tarjetas de crédito, y a partir de un número de requerimientos a las plataformas se realiza el recargo al medio de pago ingresado.

Para obtener funcionalidades gratuitas se realizan implementaciones a través de librerías no oficiales que permiten aprovechar la traducción sin incluir métodos de pago; a pesar de no ser oficiales, no tienen impedimento legal para su uso y son de libre acceso. Para *Cloud Translation* se utilizan la librerías *googletrans* y *TextBlob*; para *Translator Text* se utiliza *bing-tr*. A diferencia de trabajar directamente con las APIs oficiales, con esta metodología se debe controlar el número de requerimientos que se realizan en un intervalo de tiempo, caso contrario la plataforma de traducción bloquea la IP desde la que se realizan las peticiones.

Mediante una fase de pruebas de traducción automática se detectó que en ocasiones el traductor introduce caracteres especiales por lo que se somete al texto obtenido a una nueva fase de limpieza a través del módulo *PreCleaner*.

## 4.4. Evaluación del Corpus Construido

Al finalizar la recolección y limpieza de documentos se procederá a la evaluación del corpus obtenido. Esta se realizará a través de dos mecanismos: automáticos y manuales. Ambas evaluaciones se encuentran relacionadas, a partir de la evaluación humana se obtienen insumos que permiten trabajar a los mecanismos automáticos. Para los mecanismos automáticos se seleccionan las métricas BLEU, Distinct-1 y Distinct-2. De esta manera se logra evaluar la calidad de las traducciones y la diversidad que existe entre los textos.

### 4.4.1. Selección de los Evaluadores

Las personas que participarán como evaluadores serán escogidos de una muestra por conveniencia, según la disponibilidad de recursos. Consisten en estudiantes de último año de la Carrera de Idiomas de la Universidad de Cuenca, quienes con un nivel de inglés certificado C1, ejecutarán un rol de profesionales lingüísticos. Como lo indican Kitchenham et al. (2002), los evaluadores con estas características de nivel académico poseen los conocimientos necesarios para realizar tareas profesionales; y además, debido a que no cuentan con una alta experiencia especializada en diferentes campos se puede garantizar un nivel equiparable entre los participantes.

### 4.4.2. Selección de los Documentos de Análisis

El tamaño tentativo del corpus es de 200 mil documentos. Se elegirá una muestra aleatoria de esta población con un nivel de confianza de 95 % y un margen de error de 5 %. Con estas características, el tamaño de la muestra corresponde a 384 documentos. Debido a restricciones de tiempo para la fase de experimentación se trabajará sobre documentos con una longitud máxima de 100 palabras.

### 4.4.3. Categorías y Atributos de Análisis

Se busca evaluar la calidad de los documentos de acuerdo a dos categorías de análisis:

**Calidad de las traducciones.** Este componente se evalúa a través de medios automáticos y manuales. Los medios manuales además de entregar una apreciación profesional sobre los documentos, establecen resultados de línea base sobre los que actuará la métrica BLEU.

**Calidad de textos producidos por el modelo generacional.** La parte humana evaluará la calidad de los textos generados a un nivel semántico y sintáctico. Adicionalmente, se utilizará la métrica Distinct-1 y Distinct-2 para determinar la variedad existente dentro de los documentos que se generan.

En la Tabla 4.2 se especifican los atributos para cuantificar la calidad de los documentos de acuerdo a la categoría de evaluación, especificando el instrumento que se utilizará para lograr esto.

### 4.4.4. Metodología de Evaluación

Cada evaluador será expuesto a 30 documentos diferentes: 10 documentos en inglés originales de las fuentes de datos, 10 pares de documentos (un documento en inglés junto con su traducción al español por medios automáticos), y 10 documentos producidos por el modelo generacional. La evaluación humana toma como insumos a los requerimientos de la investigación, las categorías de análisis, las métricas, y el cuestionario. Como salida proporciona resultados de línea base sobre los que actúan

Tabla 4.2: Evaluación de los atributos de calidad según el mecanismo de implementación

| Atributo | Tipo de Componente | |
|---|---|---|
| | **Humano** | **Automático** |
| Relevancia | Cuestionario | - |
| Informatividad | Cuestionario | - |
| Fidelidad semántica | Cuestionario | - |
| Diversidad | Cuestionario | Distinct |
| Estructura sintáctica | Cuestionario | BLEU |

los mecanismos de evaluación automática para en conjunto con la evaluación manual producir los resultados que servirán al reporte final. Esto se ejemplifica dentro de la Figura 4.5.

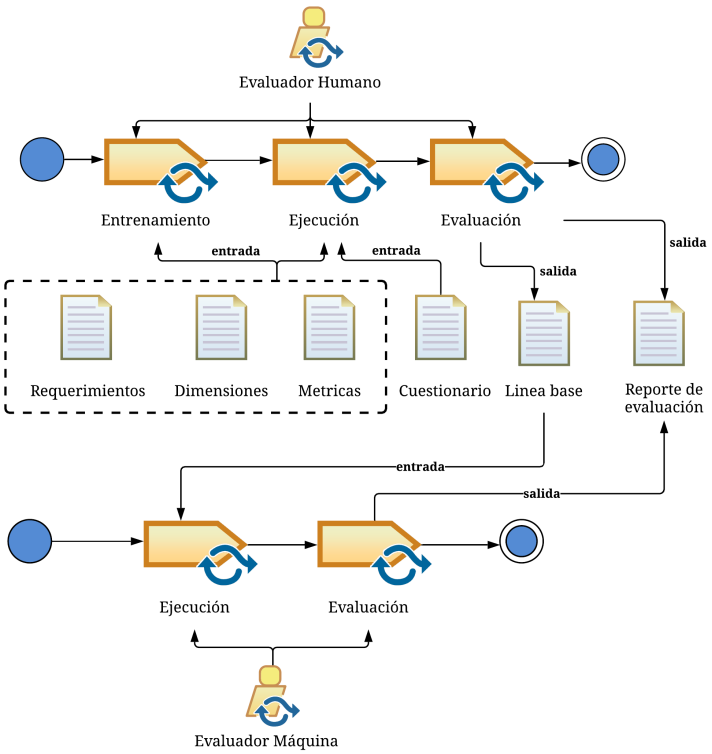


Figura 4.5: Metodología para la evaluación del corpus

Dentro del cuestionario a desarrollar, cada dimensión se cuantifica a través de preguntas cerradas. Para obtener un promedio dentro de las dimensiones de análisis se utilizará la escala de *Likert*, que constituye en una escala de conformidad (Albaum, 1997). El cuestionario se presenta en el Apéndice B; se encuentra divido en 4 secciones:

1. **Parte A - Traducción:** Obtiene las traducciones realizadas por profesionales sobre las que se obtendrá la métrica BLEU.
2. **Parte B - Traducciones automáticas:** Valora la calidad de las traducciones realizadas por los medios automáticos y la calidad de los documentos originales en inglés.
3. **Parte C - Modelo generacional:** Determina la calidad de los textos producidos por el modelo generacional.
4. **Parte D - Variedad:** Cuantifica la percepción de los evaluadores con respecto a la variedad que existió entre sus documentos de análisis.

### 4.4.5. Evaluación de la Integración de las Fuentes de Datos

Determinar con exactitud si dos fuentes de datos, especialmente cuando estos datos son textos, es una tarea de alta dificultad debido al gran número de grados de libertad que existen, y a que varios de esos grados pueden referir a una misma palabra; por ejemplo, a través de sinónimos. Ante esto se propone una combinación de un test estadístico y un análisis cualitativo para tomar una decisión informada sobre la integración de datos a realizar.

**Test Estadístico.** Se propone un test estadístico, Figura 4.6, para valorar la similitud en la distribución de palabras entre dos fuentes de datos, con base en el error *Root Mean Square Error* (RMSE). A partir de cada fuente se elabora un diccionario en el que las *llaves* corresponden al vocabulario del corpus y los *valores* corresponden a la frecuencia de aparición de la palabra dentro del corpus. Posteriormente, se normalizan (proceso de transformar a un vector a su norma unitaria) los valores del diccionario al dividirlos para la longitud del vocabulario de cada corpus. Y de este resultado, se determina el valor acumulado de RMSE; este valor corresponde al *observado* dentro del test. Si bien en este punto el valor observado no indica si dos corpus comparten la misma distribución, si puede establecer niveles de similitud entre varios corpus, al comparar los valores RMSE obtenidos.

Se establece la hipótesis nula: *El corpus A y el corpus B provienen de la misma distribución.* A continuación se procede a una fase de experimentación en donde se juntan los dos corpus a analizar (emulando una distribución compartida). De manera aleatoria se obtienen dos muestras de esta distribución compartida y se obtiene el valor RMSE que existe entre estas dos muestras. Este proceso se realiza un número $n$ de veces almacenando los resultados de cada experimento. Al finalizar la simulación del test, se calcula el valor medio y máximo del conjunto almacenado de RMSE y se lo compara con el valor observado. Si el valor observado es mayor al valor máximo registrado, quiere decir que el valor observado no es común dentro de la distribución compartida, lo que rechazaría la hipótesis nula. Por el contrario, si el valor observado se asemeja al valor promedio, es un indicativo una similitud entre las distribuciones de palabras lo que valida a la hipótesis nula.

**Valoración Cualitativa.** Como se mencionó anteriormente, debido al alto número de grados de libertad que existe en este problema, el test estadístico no es suficiente para rechazar a la hipótesis nula. Debido a esto se propone una valoración cualitativa a través de criterios personales del investigador sobre nubes de palabras generadas a partir de cada fuente de datos que conforma el corpus final. No se propone este componente con la finalidad de obtener una métrica, sino más bien una apreciación cualitativa.

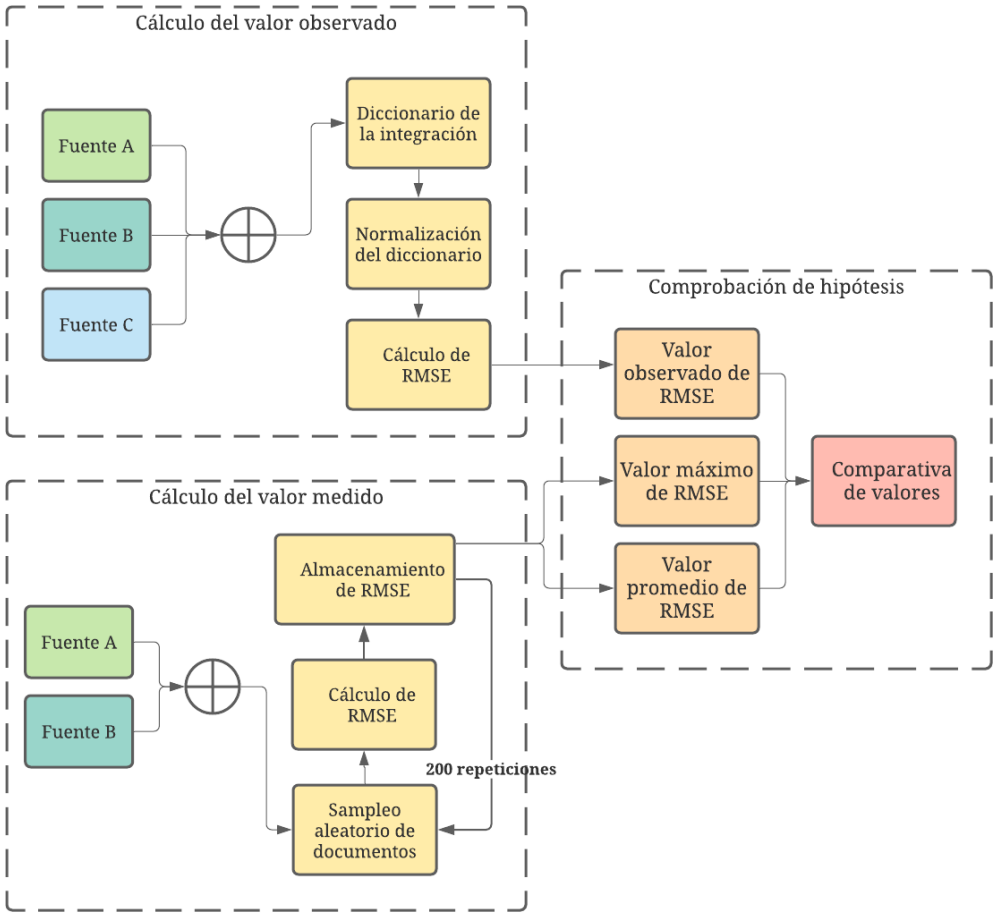


Figura 4.6: Esquema general del test estadístico para comprobar la integración de las fuentes.

## 4.5. Diseño y Desarrollo de Sistemas Soporte

A excepción del componente de evaluación humana, las tareas serán completamente automatizadas a través de software. Para esto se ha desarrollado un sistema que realiza la recolección de datos, el formateo de los mismos para adaptar los modelos de datos de las diferentes fuentes a uno unificado, la limpieza de los documentos, la traducción de los documentos en inglés y almacenamiento del corpus final. Varias de estas fases requieren de alta capacidad computacional. En la Tabla 4.3 se presenta el detalle de los equipos utilizados durante la construcción del insumo.

Tabla 4.3: Características computacionales según la etapa del proceso

| Fase | Generalidades | Detalles |
|---|---|---|
| Recolección de documentos | Un solo equipo | - |
| Limpieza | Un solo equipo | - |
| Traducción | Computación distribuida | 6 equipos |
| Evaluación | Un solo equipo | - |
| Entrenamiento del modelo | Servicios CLOUD | GPU Tesla v100, 32 GB RAM |

En la Figura 4.7 se presenta el diagrama simplificado de clases con entradas del sistema propuesto. Estas tareas requieren de alta capacidad computacional de modo que se puedan cumplir dentro de tiempos prácticos para una investigación. Existen dos clases que no se encuentran dentro de otros paquetes. Proporcionan abstracción de funcionalidades de las que pueden valerse los diferentes módulos. *FileHandler* consiste en una interfaz de comunicación con el sistema de archivos para lectura y escritura. *EventLog* consiste en una clase que registra eventos y genera reportes; de esta manera de obtiene información acerca del proceso de construcción del corpus

### 4.5.1. Paquete de Recolección de Documentos

Para cada fuente de datos se generan clases encargadas de la recolección de datos: *TwitterClient*, *MarcClient* y *AcrClient*. En estas se desarrollan las interfaces necesarias para conectarse a las fuentes de datos. Dentro de este módulo se incluye la clase *DataBuilder*, encargada de convertir los diferentes modelos de datos al modelo unificado propuesto en el presente estudio. El diagrama de clases de este módulo se presenta en la Figura 4.8. *TwitterClient* implementa los métodos encargados de solicitar documentos a la API oficial, además incorpora clases que permiten su funcionamiento. *TwitterAuthenticator* proporciona las constantes y métodos para registrar la aplicación en la API. *StdOutListener* se encarga de capturar los documentos de la API y entregarlos a los procesos consecuentes. Por último, *TweetParser* brinda un primer procesamiento al formato original de los documentos conservando los atributos requeridos para el modelo final.

### 4.5.2. Paquete de Limpieza

Proporciona dos clases principales: *PreCleaner* encargada de la primera fase de limpieza, anterior a añadir al documento dentro del corpus, y *PostCleaner* encargada de la segunda fase de limpieza, a realizar cuando el corpus ha sido generado. En la Figura 4.9 se presenta el diagrama de clases de este módulo. La clase *PreCleaner* proporciona los métodos para realizar tareas de limpieza como eliminación de caracteres especiales, remoción de urls, reemplazando de números por tokens únicos, entre otros. En cambio, la clase *PostCleaner* se encarga de eliminar documentos repetidos, corregir palabras con posibles errores ortográficos y eliminar los documentos con palabras cuya frecuencia no supere un límite

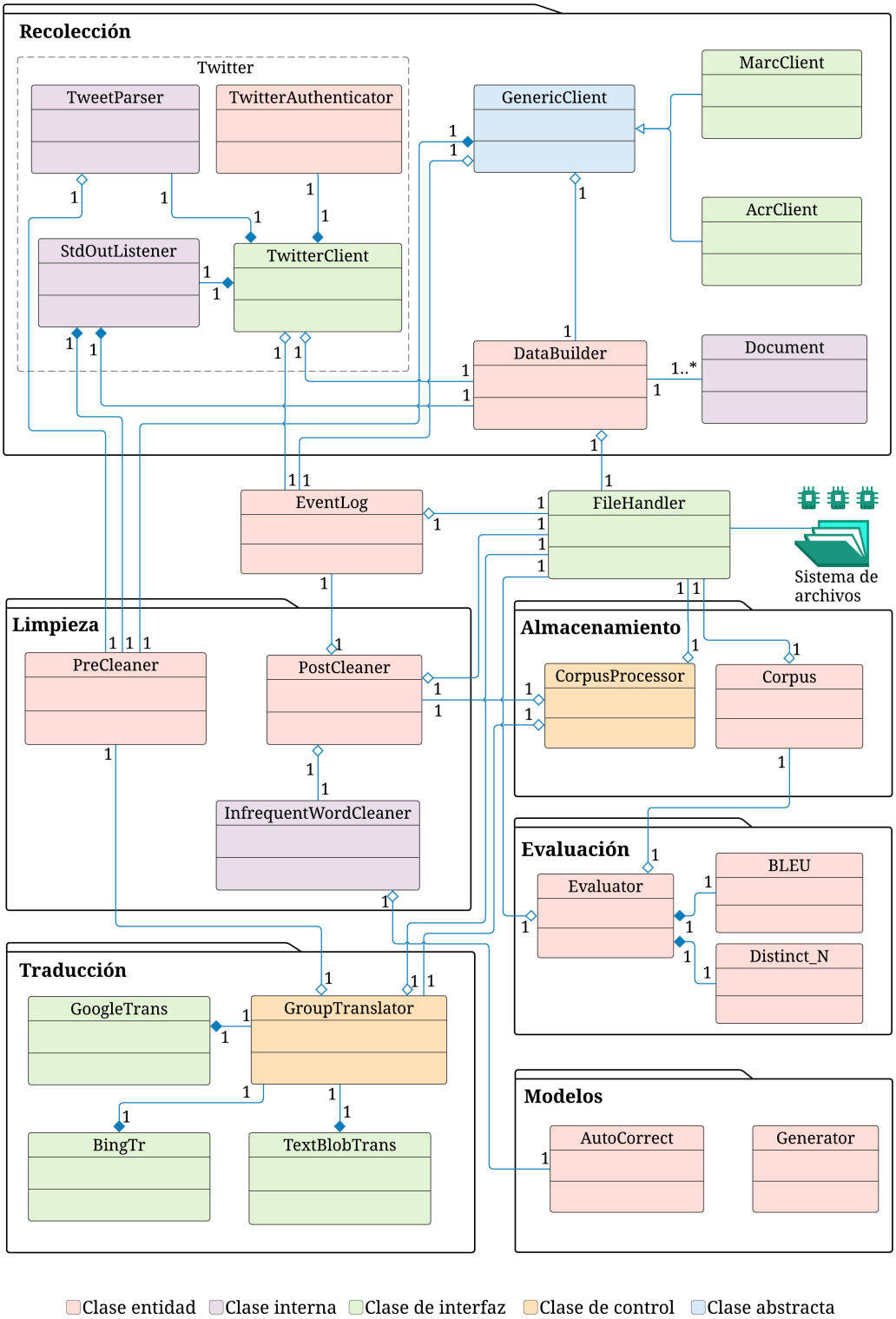


Figura 4.7: Diagrama simplificado de clases con entradas del sistema completo.

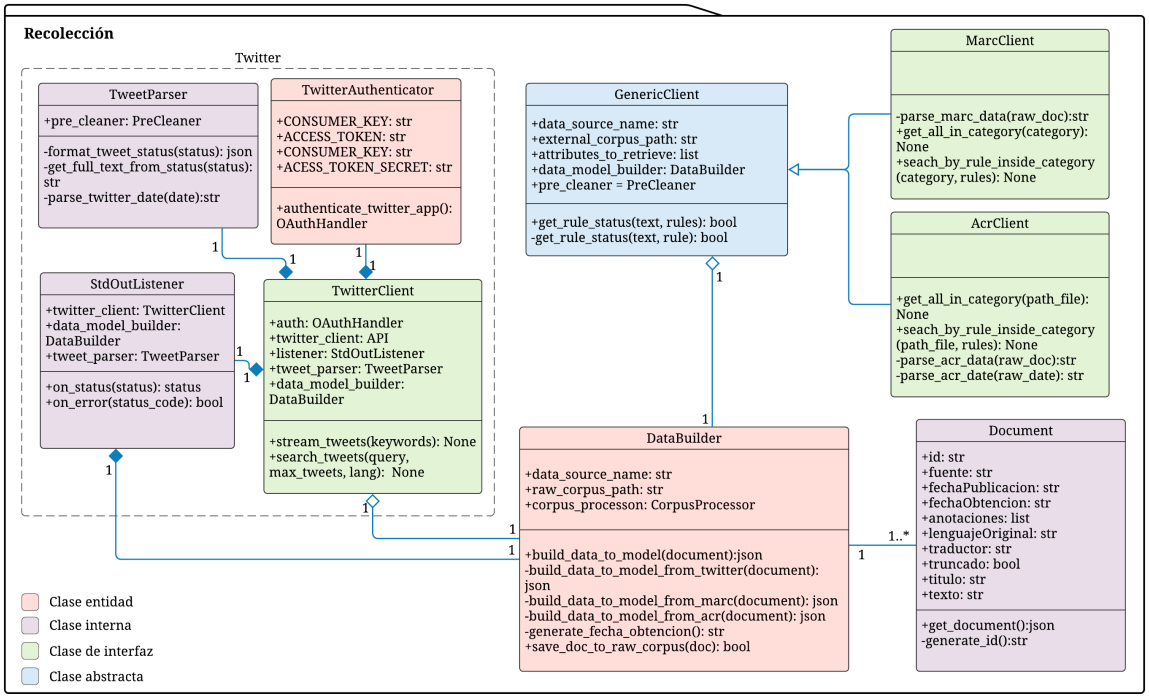


Figura 4.8: Diagrama de clases del paquete de recolección de documentos

establecido. Para corregir posibles errores ortográficos se implementa la clase *AutoCorrect*, encargada de 1) Identificar palabras con errores contrastándolas frente a un diccionario generado a partir del mismo corpus, 2) Encontrar palabras a una distancia de edición, conocida también como *edit distance*, 3) Filtrar candidatos, y 4) Reemplazar la palabra con posible error tipográfico a partir del cálculo de probabilidades.

### 4.5.3. Paquete de Almacenamiento de Documentos

Dentro del paquete de almacenamiento se proporcionan dos módulos: *CorpusProcessor* y *Corpus*. *CorpusProcessor* consiste en una clase de control. Se encarga de tomar los conjuntos de documentos almacenados en las etapas anteriores e introducirlos en la fase de traducción, post limpieza y generación de las particiones finales del corpus. *Corpus* por su parte consiste en el modulo encargado de instanciar al corpus obtenido para futuras etapas como evaluación del corpus y/o el entrenamiento del modelo generacional. El diagrama de clases de este paquete se presenta en la Figura 4.10.

### 4.5.4. Paquete de Traducción

El paquete de traducción es el encargado de tomar un documento en el idioma original y realizar su respectiva traducción al idioma objetivo; dentro de este estudio de inglés a español. Para esto trabaja con tres traductores automáticos basados en las librerías *googletrans*, *bing-tr-free*, y *textblob*; un proceso paralelizado a través de seis equipos computacionales. En la Figura 4.11 se presenta el diagrama de clases de este paquete. La clase *GroupTranslator* es de control, y se encarga de asignar un documento a los traductores disponibles. Para esta asignación se ha desarrollado un algoritmo de asignación que evite bloqueos debido a múltiples requerimientos en cortos intervalos de tiempo. Para esto luego de cada traducción establece un tiempo de pausa aleatorio tomado de una distribución normal con media

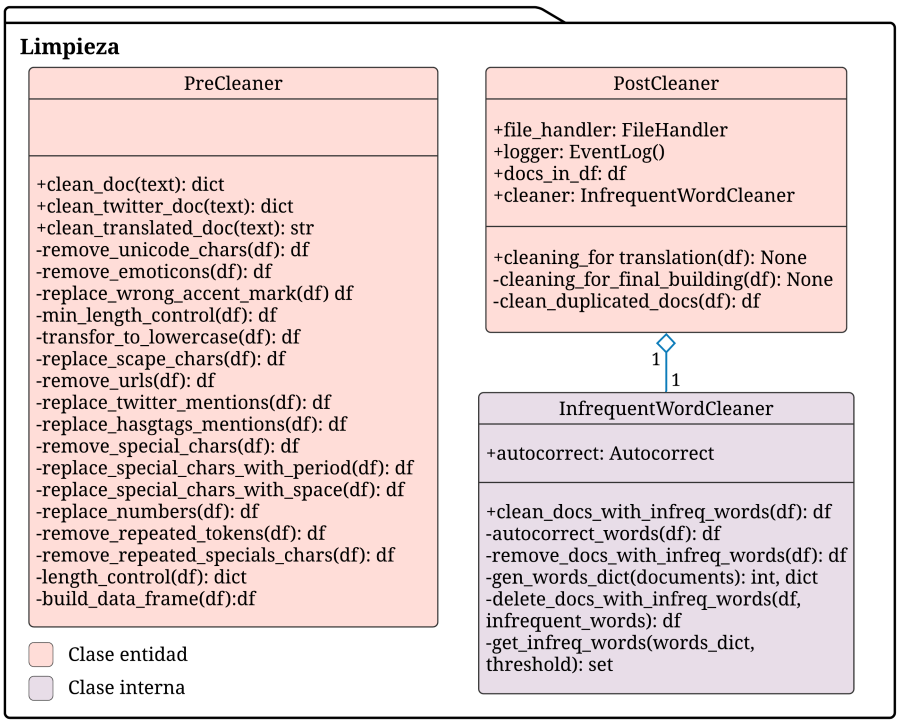


Figura 4.9: Diagrama de clases del paquete de limpieza documentos

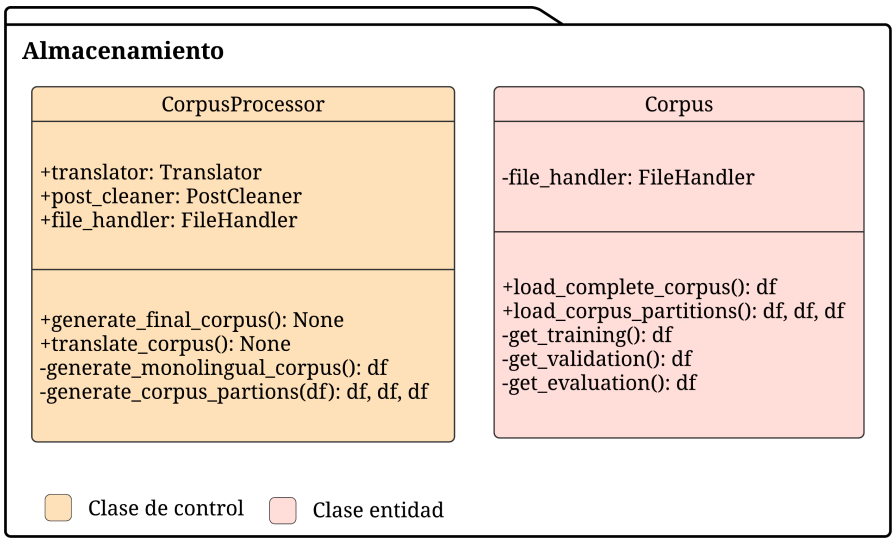


Figura 4.10: Diagrama de clases del paquete de almacenamiento documentos

2 y desviación estandar 0.1. En caso de presentarse un rechazo de solicitudes por parte de la API, el algoritmo desactiva al traductor respectivo y trabaja con los disponibles. Por su parte *GoogleTrans*, *BingTr* y *TextBlobTrans* son las clases de interfaz de conexión a las APIs de traducción encargadas netamente de traducir el documento de entrada.

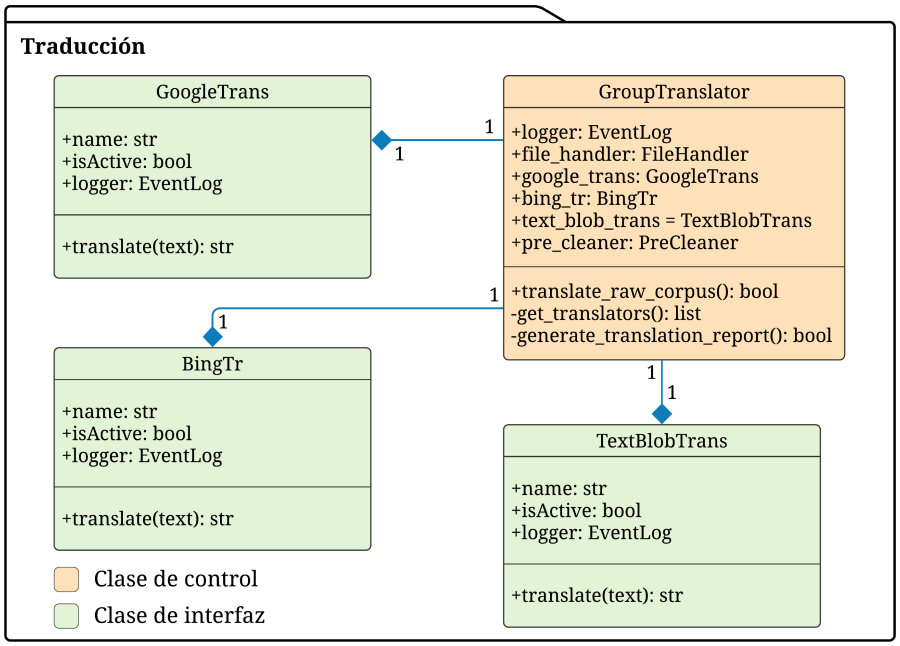


Figura 4.11: Diagrama de clases del paquete para la traducción de documentos.

### 4.5.5. Paquete de Evaluación

Los módulos dentro del paquete de evaluación no tienen dependencias con los demás paquetes de este sistema. La fase de evaluación se puede realizar de forma independiente a la construcción del corpus, además requiere de procesos manuales que entreguen insumos para realizar la evaluación automática de los documentos. *BLEU* determina la métrica *bleu* en su versión compuesta para todo el corpus de entrada, ponderando unigramas, bigramas, trigramas y cuatrigramas con una valoración de 25 % para cada combinación. Por otro lado, para *Distinct_N* se trabaja con todo el corpus a partir de unigramas y bigramas obteniendo las métricas *Distinct-1* y *Distinct-2* respectivamente. La clase *Evaluator* se encarga de determinar las métricas indicadas y proporciona métodos para formatear a los datos de entrada hacia los formatos requeridos para cada métrica. El diagrama de clases de este paquete se presenta en la Figura 4.12.

Dentro de ese paquete se almacenan los documentos de entrada para realizar las evaluaciones automáticas: hipótesis (documento de salida de una traducción automática), referencias (documento con las traducciones realizadas por personas), y predicciones (documento con los textos producidos por el modelo generacional). Para una hipótesis dada pueden existir varias referencias; en el presente estudio se utilizan dos referencias debido a la disponibilidad de recursos humanos.

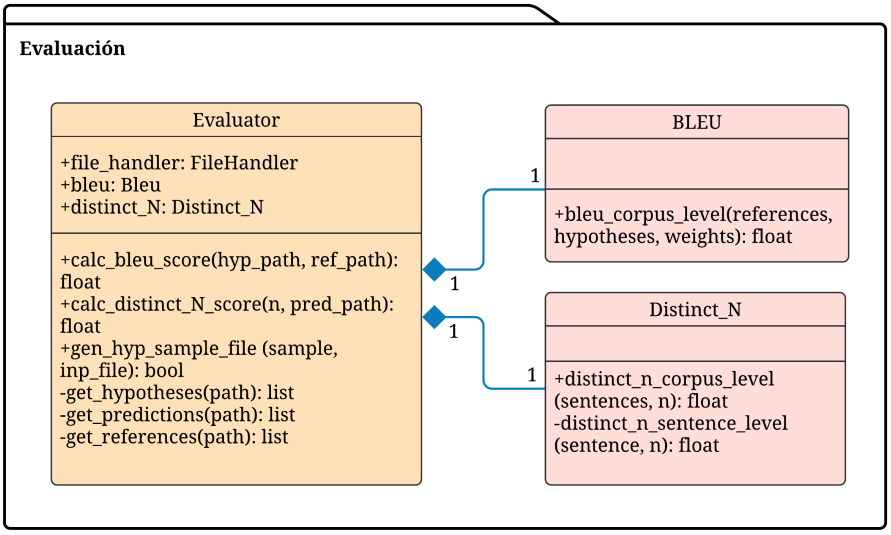


Figura 4.12: Diagrama de clases del paquete de evaluación.

### 4.5.6. Paquete de Modelos

Dentro del sistema se proporcionan dos modelos. El primero, *Autocorrect* consiste en un modelo probabilístico para auto corregir palabras del corpus cuya frecuencia no supere un límite determinado. Las correcciones se realizan a partir de una distancia de edición. Se proporciona la posibilidad de realizar la corrección a partir de dos distancias de edición; esto no fue ejecutado dentro del presente estudio debido a la alta complejidad temporal que se registró en las pruebas del sistema antes de la construcción del corpus final. Por otro lado, *Generator* se encarga de la generación automática de textos; para esto permite una etapa de entrenamiento con los datos respectivos y proporciona resultados de validación y evaluación. El diagrama de clases de este paquete se presenta en la Figura 4.13.

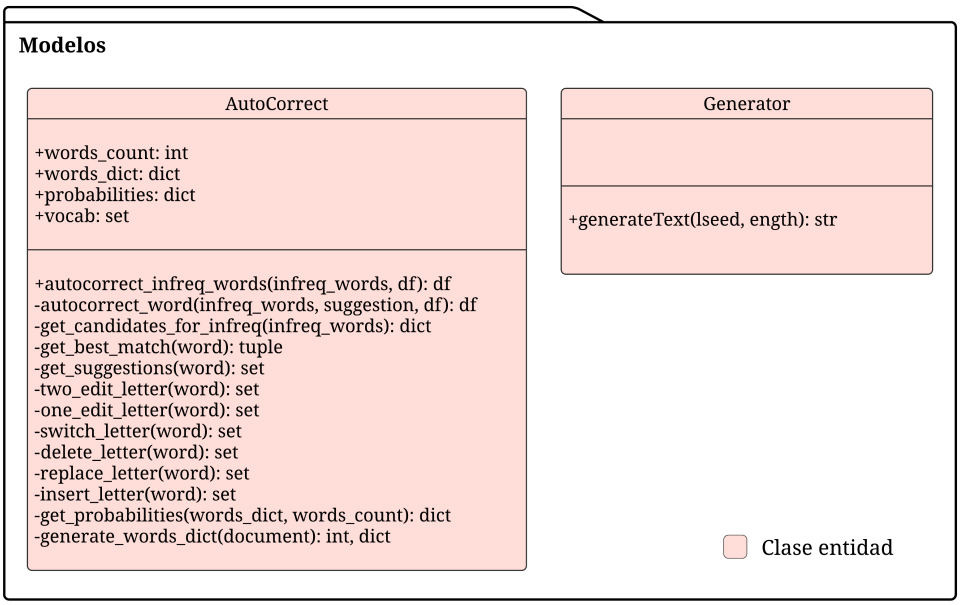


Figura 4.13: Diagrama de clases del paquete de modelos.

Capítulo 5

# Resultados y Discusión

La presente sección presenta los resultados de mayor relevancia obtenidos dentro de cada uno de los componentes desarrollados a partir de la metodología: la estructura general del corpus obtenido, los principales resultados sobre la integración de las fuentes dentro de un solo corpus, la calidad de los documentos con respecto a la relevancia y traducción. Por último, se exponen las características del modelo generacional construido junto con los resultados de línea base producidos.

## 5.1. Sistematización de Corpus Disponibles

Como se observó en la descripción del marco teórico y el estado del arte, los corpus son elaborados para objetivos específicos a partir de diversas fuentes de datos. En muchos casos a partir de un corpus se generan extensiones que incrementan sus características iniciales. En el Anexo D, se presenta el resultado de la sistematización realizada de las diferentes fuentes de textos identificadas.

## 5.2. Estructura General del Corpus

Finalizada la etapa de recolección, se obtuvo una cantidad superior a 250 mil documentos, de los cuales el 96 % se encontraban originalmente en inglés. A través del desarrollo de un sistema de apoyo se tradujeron los documentos. En la Tabla 5.1 se presenta la conformación del corpus inicial.

Tabla 5.1: Conformación del corpus inicial de acuerdo a la fuente de datos.

| **Fuente de datos** | **Cantidad de documentos** |
|---|---|
| Twitter | 5312 |
| MARC | 8948 |
| ACR | 237977 |
| Total | 252237 |

Nota: Consta de los documentos obtenidos de manera anterior a la fase de traducción.

Para la fase de post limpieza se elaboró un auto corrector basado en la métrica de distancias de edición: 1-edit. Este tuvo la tarea de generar un diccionario del corpus total y corregir las palabras que

no superaron el límite de frecuencia establecido. Posteriormente, se eliminaron los comentarios que aún poseían palabras consideradas como infrecuentes.

El resultado final consistió en un corpus con una cantidad superior a 170 mil documentos. El registro de cómo varió la cantidad de documentos luego de cada fase de procesamiento se lo detalla en la Figura 5.1. Se llama como *corpus inicial* al resultado de la recolección de documentos posterior a la fase de pre limpieza . El valor registrado en *procesamiento 1* refiere a la cantidad de documentos después de haber retirado los documentos repetidos del corpus inicial, siendo estos los que se enviaron al proceso de traducción, en esta fase existió un 0.2 % de pérdida frente a la etapa anterior. En el *procesamiento 2* se encuentran los documentos que superaron la fase de traducción, la pérdida que existió en este punto, correspondiente al 0.01 %, se debe a que en algunos casos el traductor automático devolvía valores nulos. Dentro del *procesamiento 3* se encuentran los documentos que superaron la fase de detección de lenguaje, existiendo una pérdida del 10.2 % con respecto a la etapa anterior; esto debido a que en algunos casos el traductor devolvía al mismo documento sin traducir; se presume que esto se debe a que al cumplir con cuotas de uso, la API de traducción devolvía el mismo mensaje sin traducir antes de levantar el error por cuota cumplida. Por último, el *corpus final*, corresponde a los documentos que superaron la fase de auto corrección y eliminación de documentos con palabras infrecuentes.

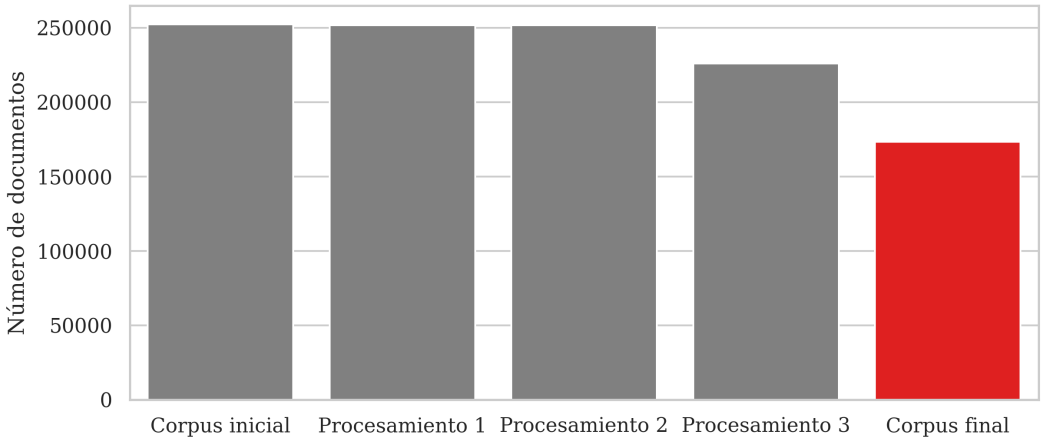


Figura 5.1: Cantidad de documentos de acuerdo a la fase de procesamiento.

Ahondando en la estructura de los documentos del corpus final, el 50 % de los documentos posee una longitud entre 26 y 35 palabras, el 25 % posee una longitud de entre 5 a 25 palabras, y el 25 % restante una longitud entre 36 y 120 palabras. El detalle de esta distribución se presenta en la Figura 5.2.

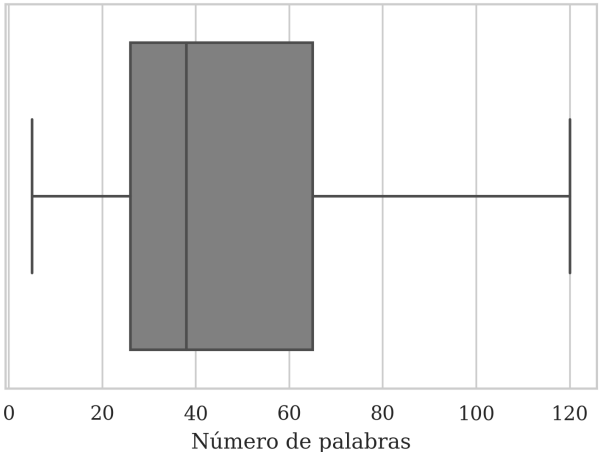


Figura 5.2: Distribución del número de palabras de los documentos del corpus final.

## 5.3. Integración de los Documentos de las Diferentes Fuentes de Datos

Cómo se mencionó dentro de la metodología, los criterios para evaluar a las distribuciones de los textos de cada fuente de datos son de carácter informativo y permiten a los investigadores tomar decisiones informadas sobre cómo integrar a los datos.

A partir del test estadístico basado en RMSE se obtienen los resultados presentados en la Tabla 5.2. Los valores observados consisten en los obtenidos al comparar las distribuciones de textos entre las diferentes fuentes; los valores máximos corresponden al valor máximo registrado dentro de una simulación repetida 300 veces; y, el valor promedio consiste en la media de los valores registrados durante la simulación. Como se aprecia, en ninguno de los casos la distribución analizada superó el test estadístico, no obstante, no se puede llegar a la conclusión de que una fuente no es parte de una distribución de datos debido a la gran cantidad de grados de libertad que engloba el problema pero sí se puede indicar que existe una mayor diferencia de similitud entre los corpus Twitter-MARC y Twitter-ACR, de la que existe entre los corpus MARC-ACR.

Tabla 5.2: RMSE obtenido en el test estadístico según la comparativa de distribuciones de textos de las fuentes.

| | **Registro** | **Twitter** | **MARC** | **ACR** |
|---|---|---|---|---|
| **Twitter** | Observado | 0 | 6.3E-04 | 4.2E-04 |
| | Máximo | 0 | 6.9E-05 | 3.8E-05 |
| | Promedio | 0 | 8.9E-05 | 5.2E-05 |
| **MARC** | Observado | 6.3E-04 | 0 | 2.9E-04 |
| | Máximo | 6.9E-05 | 0 | 2.6E-05 |
| | Promedio | 8.9E-05 | 0 | 3.7E-05 |
| **ACR** | Observado | 4.2E-04 | 2.9E-04 | 0 |
| | Máximo | 3.8E-05 | 2.6E-05 | 0 |
| | Promedio | 5.2E-05 | 3.7E-05 | 0 |

Al reflejar a los documentos de cada fuente de datos por medio de nubes de palabras se observa que, si bien existe una diferencia con respecto al uso del lenguaje, lo que explica que no se logró superar

ningún test estadístico, estos documentos aún así giran entorno al dominio textil. En la Figura 5.3 se presenta la nube de palabras del corpus ACR, aquí destacan palabras que refieren al tamaño, calidad, ajuste, comodidad, tiempo de entrega, entre otros. En la Figura 5.4, se expone la gráfica para el corpus MARC en donde las palabras que destacan son de carácter similar a las del corpus ACR. En la Figura 5.5 se presenta la nube de palabras del componente de Twitter, aquí en cambio destacan sustantivos referentes a prendas de vestir, menciones a otros usuarios y adjetivos de calidad, confort, entre otros.

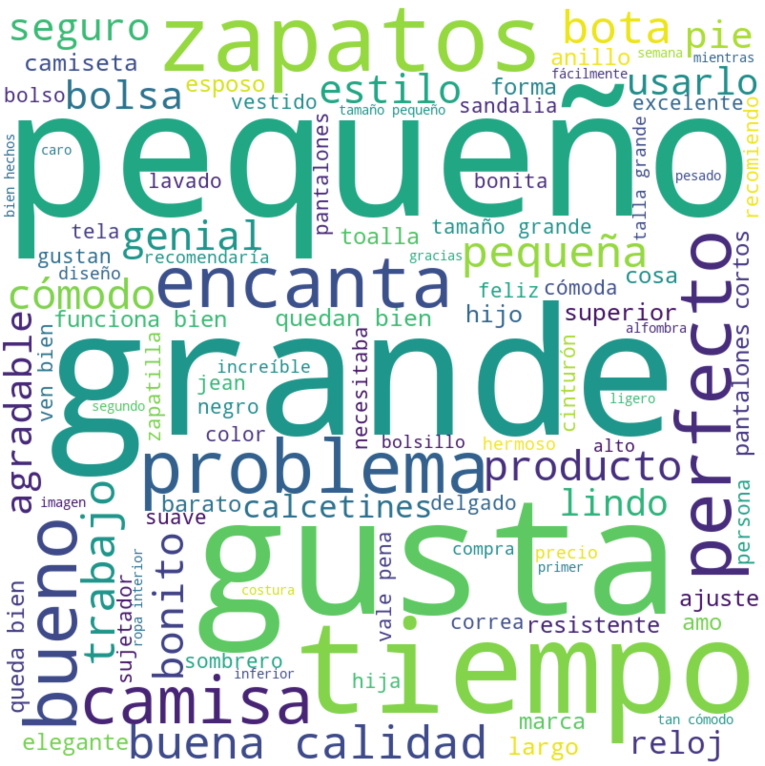


Figura 5.3: Nube de palabras - ACR.

La diferencia entre estas conformaciones se puede explicar debido a las características de cada fuente de datos. Para el caso de Amazon, los usuarios colocan su opinión dentro del producto adquirido, por lo que muchos ya no incluyen sustantivos que identifican a la prenda de referencia, pasando a un tono más descriptivo. En cambio en Twitter, al ser una plataforma de opinión general, los usuarios especifican la prenda a la que refiere su opinión; aquí destaca el uso de identificadores a otras cuentas de usuario por lo que se identifica que los usuarios de esta plataforma no buscan dar opiniones aisladas que hagan eco en usuarios en general, sino más bien son opiniones que buscan ser escuchadas por usuarios específicos.


Figura 5.4: Nube de palabras - MARC.


Figura 5.5: Nube de palabras - Twitter.

## 5.4. Dominio del Corpus

Se valora si los textos obtenidos al haber implementado el framework corresponden al dominio especificado. En este caso, 59 % de los documentos se encontraban claramente dentro del dominio establecido. El 17 % de los documentos, de manera, clara no corresponden al dominio establecido. Y, el restante 24 % son valorados en una posición intermedia. Al ahondar en estos últimos documentos, se identifica que consisten en textos que tienen un carácter genérico, y que su contenido pudiera ser considerado de cualquier dominio, como por ejemplo el texto "Me encantó, lo volvería a comprar sin pensarlo dos veces". En la Figura 5.6 se presenta el detalle de los resultados obtenidos en este componente.

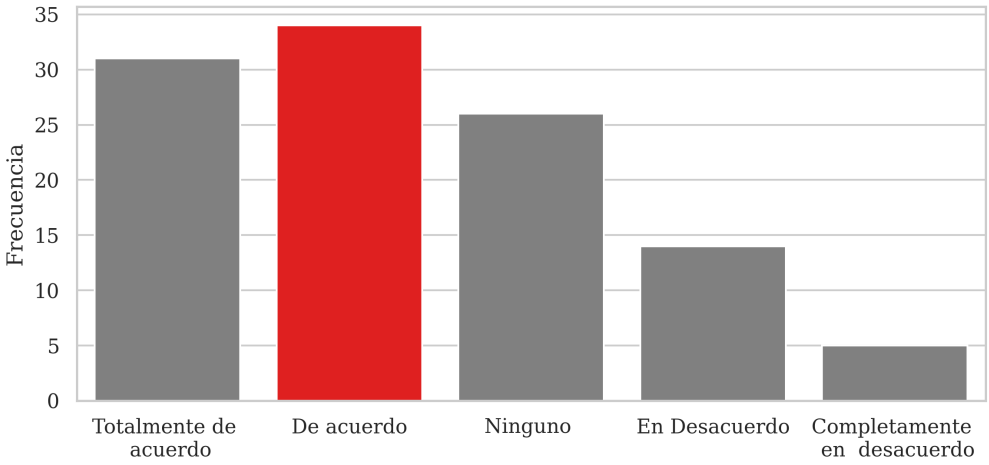


Figura 5.6: Dominio adecuado de los documentos recolectados, resaltando la categoría de mayor frecuencia.

## 5.5. Calidad de los documentos traducidos

A partir del corpus final se procedió a la fase de evaluación de las traducciones por medios automáticos. Para esto se utilizó la métrica BLEU en una configuración de unigramas, bigramas, trigramas, cuatrigramas, y en una configuración ponderada en donde a cada componente de n-gramas se le asignó un peso de 0.25. Estos resultados se sintetizan en la Figura 5.7.

Si bien el score disminuye conforme se aumenta el orden de los n-gramas, en todas las configuraciones esta supera al valor de 0.5. De acuerdo a Lavie (2011), un puntaje superior a 0.5 refleja una buena y fluida traducción. Si bien BLEU es una métrica ampliamente utilizada, y además considerando que dentro de esta investigación este score es alto, se deben realizar valoraciones adicionales; tal como lo identifica Vogel et al. (2018), esta métrica presenta limitaciones al basarse únicamente en la precisión de n-gramas. Es así que, para brindar mayor robustez a los resultados se implementó la evaluación humana referida en el framework.

Con respecto a la calidad de la traducción de los documentos obtenidos, únicamente el 4 % se considera como que no posee una traducción adecuada frente al texto original, el 72 % considera que sí posee una calidad adecuada, mientras que el restante 24 % mantienen una postura neutral frente a la traducción. Este último valor se presenta debido a que los textos de entrada, ya presentaban

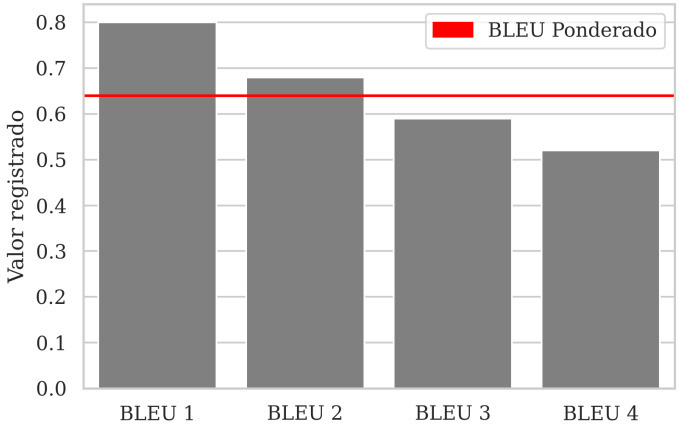


Figura 5.7: Métrica BLEU en diferentes configuraciones

dificultades para entenderlos debido a errores en su escritura, frente a esto los evaluadores no tenían una postura clara sobre si la traducción automática era adecuada o no. Los valores netos de estos resultados se presentan en la Figura 5.8.

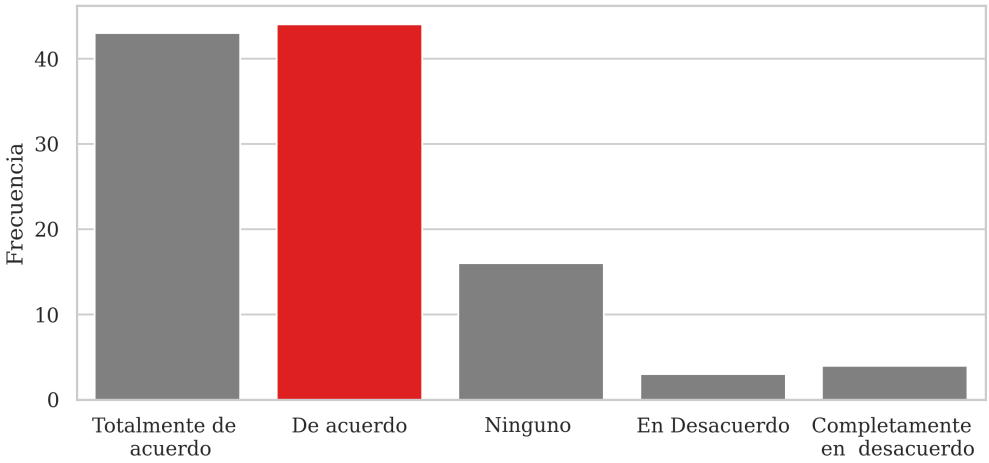


Figura 5.8: Calidad de traducción adecuada, resaltando la categoría de mayor frecuencia.

La valoración obtenida con respecto a la conformidad de los evaluadores con respecto a la sintaxis y estrutura morfológica tanto de los documentos de entrada, originales en inglés, como a los documentos de salida, producidos por traductores automáticos en español, es adecuada.

En la Figura 5.9 se presentan las distribuciones de los documentos según la dimensión de análisis. El Rango inter-cuartil para los documentos de entrada es de 1 unidad en la escala analizada y se encuentra entre 4 y 5. Entonces el 75 % de estos documentos se encuentran en un nivel de conformidad entre “De Acuerdo” y “Totalmente de Acuerdo” frente a las dimensiones establecidas. En cambio, los documentos de salida poseen un rango inter-cuartil de 2 unidades de escala entre los niveles de conformidad “Ni de Acuerdo, ni en Desacuerdo” y “Totalmente de Acuerdo”, representando el 75 % de los datos dentro de estos niveles. Ambos tipos de documentos, presentan una distribución asimétrica positiva que se traduce en que las valoraciones se encuentran sesgadas hacia un nivel de conformidad

alto con respecto a la calidad de las dimensiones analizadas.

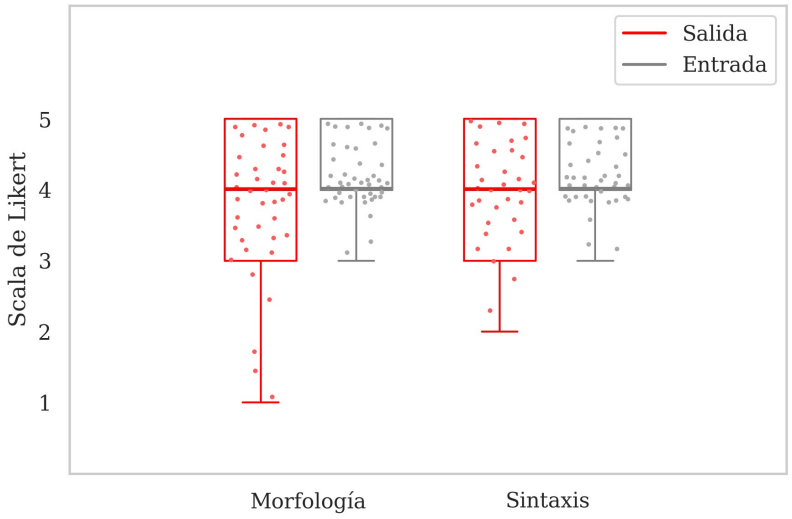


Figura 5.9: Comparativa entre la valoración de la sintaxis y estructura morfológica de los textos de entrada y salida.

## 5.6. Modelo Generacional

Se realizaron diferentes configuraciones sobre modelos generacionales basados en arquitecturas LSTM y GRU. En la Tabla 5.3 se presentan una comparativa entre las diferentes configuraciones. Los mejores resultados se obtuvieron en la iteración 7.

Tabla 5.3: Comparativa entre diferentes configuraciones de modelos generacionales.

| | **Iteración del modelo** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Características** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Secuencia de entrada | 100 | 100 | 100 | 100 | 80 | 100 | **100** |
| Dimensión capa embedding | 256 | 256 | 256 | 256 | 256 | 256 | **256** |
| Capas ocultas | 7 | 1 | 7 | 9 | 7 | 2 | **2** |
| Tipo de neurona | LSTM | GRU | LSTM | GRU | GRU | GRU | **GRU** |
| Unidades por capa | 512 | 1024 | 1024 | 1024 | 512 | 1024 | **512** |
| Número de parámetros | >10M | >3M | >10M | >15M | >10M | >10M | **>3M** |
| Épocas de entrenamiento | 10 | 20 | 20 | 50 | 50 | 50 | **70** |
| Función de pérdida | Adam | Adam | Adam | SGD | SGD | SGD | **SGD** |
| Tiempo de entrenamiento | 1h | 1h | 3h | 10h | 8h | 5h | **20h** |

La arquitectura del modelo con mejores resultados se presenta en la Figura 5.10. Fue entrenado a nivel de caracteres, debido a la disponibilidad computacional. Frente a los diferentes tipos de

optimizadores se realizaron pruebas con ADAM y SGD. Con el primero los tiempos de convergencia eran considerablemente menores; no obstante, al momento de comparar los textos producidos frente a un modelo optimizado con SGD, se podía apreciar una deficiencia considerable en los resultados producidos. Debido a esto para la arquitectura final se seleccionó al optimizador SGD con un factor de aprendizaje de 0.01 y entrenado por 70 épocas durante 20 horas en el equipo computacional descrito en la metodología. Se debe tener en consideración que no se pudo entrenar modelos de mayor capacidad, ya que bajo las prestaciones de equipos disponibles los tiempos de entrenamiento no podían superar las 24 horas. Al superar ese tiempo el equipo se restablecía (Políticas de Uso de Google Colab PRO), pérdiendo los avances.

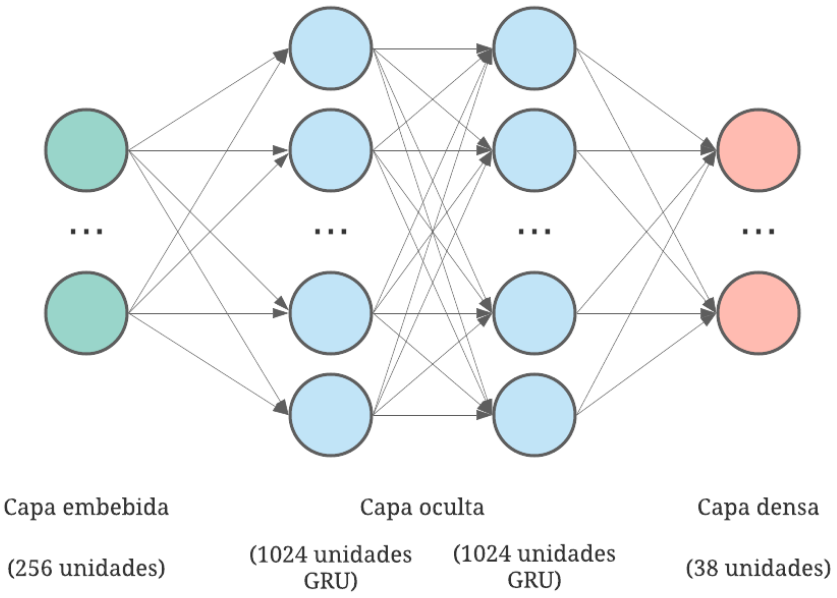


Figura 5.10: Arquitectura del modelo generacional seleccionado.

Por las características de los datos del problema, un conjunto discretos que entrega vectores esparcidos, se seleccionó la función de pérdida *Sparse Categorical Crossentropy*, basando al entrenamiento en la métrica de *accuracy*. Los resultados de estas funciones dentro de cada época se presentan en la Figura 5.11. Al finalizar el entrenamiento, el valor de exactitud alcanzado fue de 0.7007 y el valor de la función de pérdida 0.941.

La valoración sobre la calidad de los textos producidos por el modelos generacional fue desarrollada por el componente humano. Para esto, profesionales lingüísticos valoraron un conjunto de 110 documentos con longitud máxima de 30 palabras cada uno. En este punto se buscó proporcionar resultados de base para que futuras investigaciones puedan ahondar esfuerzos y mejorar los resultados obtenidos.

Con respecto a la relevancia de la temática de los documentos obtenidos frente al tema de estudio, opiniones de productos textiles, y a la informatividad de estos, los resultados se presentan en la Figura 5.12. El nivel de relevancia se distribuye de forma simétrica, de este modo, el 50 % de los documentos son considerados relevantes al tema de estudio, lo que indica que el modelo es capaz de escribir textos que buscan emitir un comentario de productos textiles. En cuanto, a la informatividad de los documentos, referida a si el texto generado entrega información útil dentro del comentario o es una serie de palabras aleatorias, el 25 % de los textos entregan un nivel adecuado de información.

Los resultados obtenidos dentro de las dimensiones de calidad de sintaxis y estructura morfológica

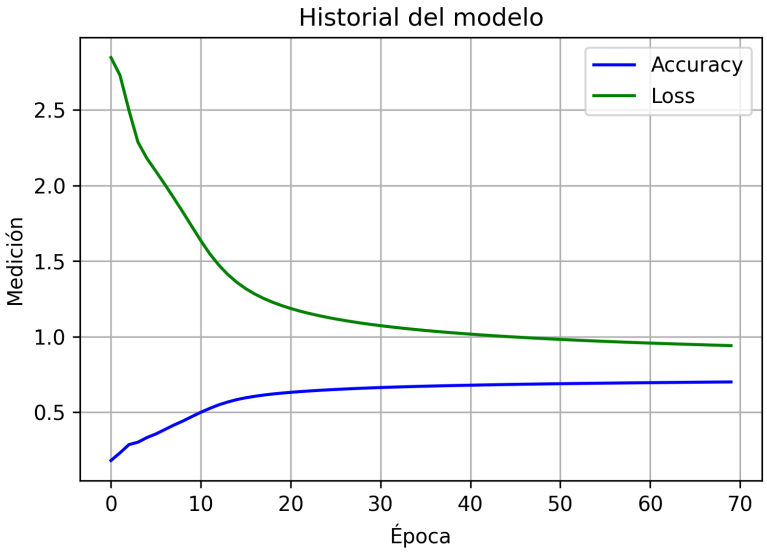


Figura 5.11: Historial del entrenamiento del modelo para los valores de pérdida y exactitud.

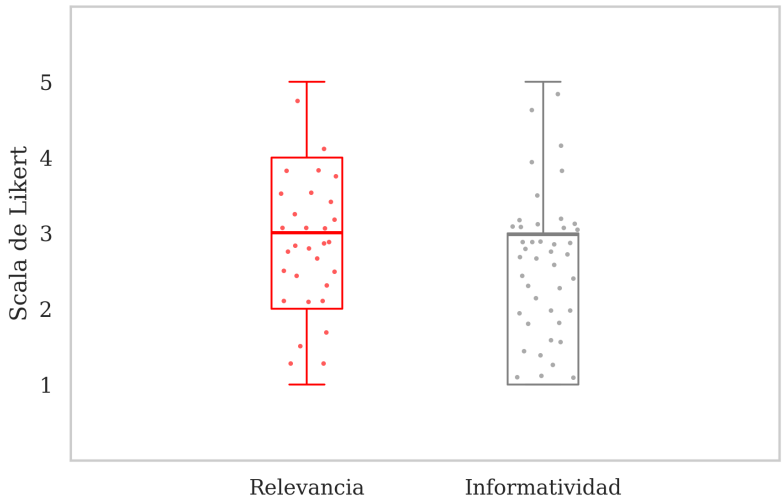


Figura 5.12: Relevancia e informatividad de los documentos producidos.

de los textos los resultados se muestran en la Figura 5.13. En general, el 25 % de los documentos presentan características sintácticas y morfológicas adecuadas. El modelo obtenido no fue capaz de aprender de forma adecuada las reglas sintácticas y morfológicas que rigen el uso del lenguaje.

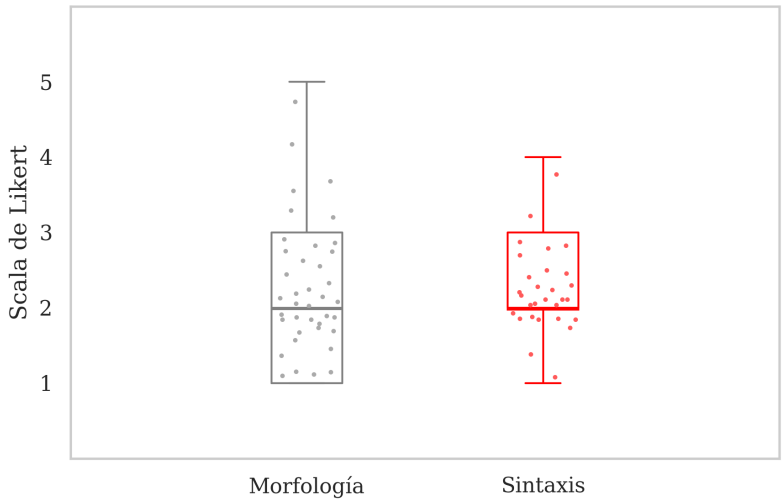


Figura 5.13: Morfología y sintaxis de los documentos producidos.

Habiendo descrito los resultados principales, se presentan varios textos generados en los que se puede apreciar además de mantener el dominio requerido, las características sintácticas y morfológicas valoradas por los expertos.

> ***Pijama*** *bastante fácil de comprar un mes el material es un poco justo pero demasiado grande y se lava y bien para algunos de estos zapatos tengo estos calcetines son de cinco problemas bastante cómodo.*
> ***Camisa*** *se sentía perfecta para mis pies compré este collar y en el primer par de pantalones cortos como en la casa estos son súper bien el cordón para el producto simple.*
> ***Buzo*** *correcto estos jeans son grandes los sombreros se rasgarán no son para los días para viajar en la parte delantera de la correa se siente muy bien y es muy delgada es cómodo.*
> ***Abrigo*** *de verano excelente comprado solo perfecto para los cordones para algunas caracteristas ventanas muy bonitas y compré el producto este es el ajuste perfecto en el material parece totalmente está limpio.*

Capítulo 6

# Conclusiones y Recomendaciones

## 6.1. Conclusiones

El objetivo de esta investigación consistió en construir un corpus en español de alta calidad y gran escala dentro del dominio de productos textiles, con capacidad de alimentar a modelos de Aprendizaje Profundo. Para esto se desarrolló una metodología que describe detalladamente cada una de las fases para la construcción de este insumo.

Cuatro fueron los resultados principales obtenidos dentro de la investigación: 1) Un corpus en español con más de 170 mil documentos que obtuvo buenos resultados dentro de la fase de evaluación humana y evaluación automática, 2) Un sistema computacional que automatizó completamente la construcción del corpus, desde la recolección de los documentos hasta su evaluación, 3) una metodología altamente escalable y adaptable a diferentes dominios e idiomas, y 4) resultados de línea base de un modelo generacional que sirven como punto de referencia para futuras investigaciones dentro de la generación automática de textos dentro del dominio textil.

Un punto clave dentro de la investigación consistió en la traducción de documentos a partir del idioma inglés; debido al gran volumen de datos disponibles en este idioma. Si bien los documentos traducidos presentan una mayor variabilidad de su calidad frente a los documentos originales, estos siguen estando sesgados hacia valoraciones altas de calidad, equiparándose a los textos originales. También se debe considerar que una parte de los textos de entrada presentan valoraciones de baja calidad, siendo una muestra de que un documento producido por personas no es garantía de una sintaxis y estructura morfológica adecuada.

Determinar con exactitud si se pueden integrar fuentes de datos es una tarea de alta dificultad debido al gran número de grados de libertad que existen dentro de un corpus. Sin embargo, a través del test estadístico propuesto, y el análisis cualitativo de las las nubes de palabras de cada fuente se puede tomar una decisión informada sobre esta integración.

No basta con determinar un diseño metodológico para construir insumos con estas características, adicionalmente se debe disponer de una alta capacidad computacional para que la ejecución de los procesos puedan realizarse en tiempos prácticos para una investigación. Características como computación distribuida y tarjetas GPU de altas prestaciones son requeridas. Según las características del sistema, sí se buscase desarrollar la construcción del corpus obtenido dentro de un solo computador

de bajas características, los tiempos de recolección y traducción se pueden extender a 60 días. Al pasar a un equipo con mejores características, se reducen los tiempos de los procesos de limpieza, aminorando la complejidad temporal de este problema. Para este estudio se trabajó con un cluster de equipos de gama media logrando ejecutar la tarea de recolección y limpieza en 10 días. Además, para la generación de textos, la utilización de tarjetas gráficas de alta gama redujeron los tiempos de entrenamiento a una décima parte de lo que hubiera tomado un entrenamiento sin GPU.

## 6.2. Recomendaciones

La construcción de un corpus de gran escala es una tarea ardua en donde los procesamientos de las diferentes etapas pueden tomar desde un par de minutos hasta varios días. Desde un punto de vista práctico, en el que la investigación pueda desarrollarse dentro de tiempos realistas, se debe contar con equipos computacionales de alto rendimiento, que permitan paralelizar tareas o integrar una GPU en el caso de que se busque entrenar a modelos de aprendizaje. Estos equipos pueden reducir las tareas a una mínima fracción frente a equipos que no sean de alto rendimiento.

En cuanto a especificaciones de hardware, para la construcción del corpus se recomiendan equipos de al menos 4GB de RAM, con procesadores con capacidad de paralelizar las tareas en varios núcleos. Por otro lado, para la generación de textos se recomiendan equipos con al menos 32GB de RAM, y tarjetas gráficas de 6GB; estas características no son suficientes para modelos robustos, sin embargo, permiten obtener resultados de análisis exploratorios en tiempos y costos prácticos.

## 6.3. Trabajos futuros

Se detallan a continuación posibles líneas de trabajo que se lograron vislumbrar durante la ejecución de la presente investigación:

- Migración del sistema desarrollado a tecnologías de Big Data como por ejemplo Spark.
- Construcción de un corpus masivo, a partir del cual se pueda valorar si las fuentes de datos superan el test estadístico desarrollado, a modo de demostrar que una fuente puede ser considerada de la misma distribución que otra fuente.
- Entrenamiento de modelos generacionales de arquitecturas complejas sobre el corpus generado. Por ejemplo, modelos con 100M de parámetros, o modelos que se componen de una combinación de redes LSTM, VAEs, entre otros.
- Implementación de la metodología descrita para otros lenguajes y dominios.

ANEXO A

# Modelos Completos de las Fuentes de Datos

## A.1. ACR

reviewerID": .A1KLRMWW2FWPL4", .asin": "0000031887", reviewerName": .Amazon Customer c̈ameramom", "helpful": [0, 0], reviewText": "This is a great tutu and at a really great price. It doesn't look cheap at all. I'm so glad I looked on Amazon and found such an affordable tutu that isn't made poorly. A++", .overall": 5.0, "summary": "Great tutu- not cheaply made", ünixReviewTime": 1297468800, reviewTime": "02 12, 2011

METADATA .asin": "0000031852", "title": "Girls Ballet Tutu Zebra Hot Pink", "price": 3.17, ïmUrl": "http:", related": .also_bought": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O", "0000031909"] , "salesRank": "Toys Games": 211836, "brand": Çoxlures", çategories": [["Sports & Outdoors", .other Sports", "Dance"]]

## A.2. MARC

review_id": .es_0491108", "product_id": "product_es_0296024", reviewer_id": reviewer_e_0999081", "stars": "1", review_body": "Nada bueno se me fue ka pantalla en menos de 8 meses y no he recibido respuesta del fabricante", review_title": "television Nevir", "language": .es", "product_category": .electronics

## A.3. Twitter

( 'created$_at$' :' $ThuApr2915:00:29+00002021$',' $full\_text$' :' $NuevaresolucióndelCOENacional...$' ... lang=és'possibly_sensitive=False ... )

Status(_api=<tweepy.api.API object at 0x019DFEF0>, _json={'created_at': 'Thu Apr 29 17:16:37 +0000 2021', íd': 1387818112532287489, íd_str': '1387818112532287489', 'full_text': Ínvestigación sobre pérdida de piezas patrimoniales del Museo Pumapungo. Cuenca Azuay Ecuador CCIOnline CCINoticias Noticias https://t.co/b71z6nyAD5', 'truncated': False, 'display_text_range': [0, 131], éntities': 'hashtags': ['text': 'Cuenca', índices': [74, 81], 'text': Ázuay', índices': [82, 88], 'text': Écuador', índices': [89, 97], 'text': 'CCIOnline', índices': [98, 108], 'text': 'CCINoticias', índices': [109, 121], 'text': Ñoticias',

índices': [122, 131]], 'symbols': [], úser_mentions': [], úrls': [], 'media': [íd': 1387818070417330177, íd_str': '1387818070417330177', índices': [132, 155], 'media_url': 'http:', 'media_uRL_https': 'https', úrl': 'https:', 'display_url': 'pic.twitter.com/b71z6nyAD5', 'thumb': 'w': 150, 'h': 150, 'resize': 'crop
, 'medium': 'w': 696, 'h': 464, 'resize': 'fit
, 'small': 'w': 680, 'h': 453, 'resize': 'fit
, 'large': 'w': 696, 'h': 464, 'resize': 'fit
], 'source': '<a href="https://mobile.twitter.comrel="nofollow»Twitter Web App</a>', , geo=None, coordinates=None, place=None, contributors=None, is_quote_status=False, retweet_count=3, favorite_count=0, favorited=False, retweeted=False, possibly_sensitive=False, lang=és')

ANEXO B

# Cuestionario de Evaluación por Componente Humano

## B.1. CUESTIONARIO

### B.1.1. Parte A - Traducción

Antecedentes: Se entregará a cada participante 10 textos obtenidos de forma aleatoria de las fuentes de documentos en inglés. A partir de esta valoración se obtiene la métrica BLEU.

Indicaciones: Para cada uno de los textos proporcionados en inglés, realice su respectiva traducción al español.

### B.1.2. Parte B – Traducciones automáticas

Antecedentes: Se entregará a cada participante 10 pares de documentos. La versión original en inglés con su respectiva traducción realizada por medios automáticos. Indicaciones: Asignar una valoración en la escala de 1 a 5 para los siguientes enunciados.

- La traducción realizada es adecuada.
- El texto de salida guarda relación semántica con el documento original.
- El texto de salida posee una morfología adecuada.
- El texto de salida posee una estructura sintáctica adecuada.
- El texto de entrada posee una morfología adecuada.
- El texto de entrada posee una estructura sintáctica adecuada.

### B.1.3. Parte C – Modelo generacional

Antecedentes: Se entregará a cada participante 10 documentos obtenidos a partir del modelo generacional.

Indicaciones: Asignar una valoración en la escala de 1 a 5 para los siguientes enunciados.

- El texto es relevante al tema de estudio.
- El texto es informativo, independientemente del tema de estudio.
- El texto posee una morfología adecuada.
- El texto posee una estructura sintáctica adecuada.

### B.1.4. Parte D - Variedad

Antecedentes: Se solicitará a cada participante responder a una pregunta sobre cada parte del cuestionario.

Indicaciones: Asignar una valoración en la escala de 1 a 5 para los siguientes enunciados.

- Los textos entregados en la Parte B presentan variedad entre ellos.
- Los textos entregados en la Parte C presentan variedad entre ellos.

ANEXO C

# Reglas y Cadenas de Búsqueda

## C.1. Twitter

twitter_rules = [ outdoor cover"(bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "b̈ed covers̈(bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "f̈urniture cover̈(bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(b̈ed sheetÖR b̈ed sheets)̈ (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(curtain OR curtains) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(b̈ed sheetÖR b̈ed sheets)̈ (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(tablecloth OR tableclothes) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(towel OR towels) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(carpet OR carpets) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(rug OR rugs) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(apron OR aprons) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(uniform OR uniforms) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(quilt OR quilts) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(blanket OR blankets) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(coat OR coats) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(sock OR socks) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(shirt OR shirt) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(shirt OR shirts OR tshirt OR t-shirt) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(hood OR hoods) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)",

"(jacket OR jackets) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(vest OR vests) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(sweater OR sweaters) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(skirt OR skirts) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(cap OR caps) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(glove OR gloves) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(pant OR pants) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(trousers) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "shorts (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(hoodie OR hoodies) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(pijama OR pajama) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "p̈olo shirt(̈bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "sleepwear (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", ünderwear (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(sweatshirt OR sweatshirts) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(suit OR suits) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(dress OR dresses) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(shoe OR shoes) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(f̈ace mask)̈ (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(panty OR panties) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", ünderpants (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)", "(outfit OR fashion OR clothing OR clothes) (bought OR buy OR adquire OR got OR like) (quality OR comfort OR nice OR beautiful OR colorfull OR wonderfull)"]

## C.2. ACR

acr_rules = [ "(outdoor covers | outdoor cover) () ()", "(bed covers | bed cover) () ()", "(furniture covers | furniture cover) () ()", "(bed sheets | bed sheet) () ()", "(curtains | curtain) () ()", "(tablecloth | tableclothes) () ()", "(towels | towel) () ()", "(carpet | carpets) () ()", "(rug | rugs) () ()", "(apron | aprons) () ()", "(uniform | uniforms) () ()", "(quilt | quilts) () ()", "(blanket | blankets) () ()", "(coat | coats) () ()", "(sock | socks) () ()", "(shirt | shirts) () ()", "(t-shirt | tshirt | t-shirts | tshirts) () ()", "(hood | hoods) () ()", "(jacket | jackets) () ()", "(vests) () ()", "(sweater | sweaters) () ()", "(skirt | skirts) () ()", "(cap | caps) () ()", "(glove | gloves) () ()", "(hoodie | hoodies) () ()", "(pants | pant) () ()", "(trousers) () ()", "(shorts) () ()", "(pijama | pajama) () ()", "(polo shirt) () ()", "(sleepwear) () ()", "(underwear) () ()", "(sweatshirt | sweatshirts) ()( )", "(suit | suits) () ()", "(dress | dresses) () ()", "(shoe | shoes) () ()", "(face mask) () ()", "(panty | panties) () ()", "(underpants) () ()", "() (fashion) ()"]

## C.3. MARC

marc_rules= [ "(cobertores para exterior | cobertor para exterior) () ()", "(cobertores de cama | cobertor de cama) () ()", "(cobertores para muebles | cobertor para mueble | cobertor para muebles) () ()", "(sábanas | sábana) () ()", "(cortinas | cortina) () ()", "(mantel | manteles) () ()", "(toallas|toalla)()()", "(alfombra|alfombras)()()", "(tapetes|tapete)()()", "(mandiles|mandil)()()", "(uniformes|uniforme)()()", "(edredón|edredon|edredones)()()", "(cobija|cobijas)()()", "(colcha|colchas)()()", "(abrigo|abrigos)()()", "(buzo|buzos)()()", "(calcetin|calcetines|calcetín)()()", "(camisa | camisas) ()()", "(camiseta | camisetas) () ()", "(capucha | capuchas) () ()", "(casaca | casacas) () ()", "(chaleco | chalecos) () ()", "(chaqueta | chaquetas) () ()", "(chompa | chompas)() ()", "(falda | faldas)() ()", "(gorra | gorras)() ()", "(guante | guantes) ()()", "(hoodie | hoodies) ()()", "(pantalon |pantalón | pantalones)()()", "(pantaloneta|pantalonetas)()()", "(pijama | pijamas)()()", "(polo|polos)()()", "(ropa de dormir)()()", "(ropa interior)()()", "(sudadera|sudaderas)()()", "(terno|ternos)()()", "(vestido|vestidos)()()", "(zapato|zapatos)()()", "(mascarilla|mascarillas)()()", "(braga|bragas)()()", "(calzoncillos|calzoncillo)()()", "()(fashion)()", ]

ANEXO D

# Sistematización de corpus disponibles

Tabla D.1: Sistematización de corpus disponibles.

| **Nombre** | **Autor** | **Num. Lenguajes** | **Documentos por lenguaje** | **Licencia** | **Objetivo del corpus** | **Principal fuente de datos** |
|---|---|---|---|---|---|---|
| Europarl | (Koehn, 2005) | 11 | Varía desde 200 mil a 2 millones | CC0 | Tareas de traducción por máquina. | Actas del parlamentos europeo |
| SNLI | (Bowman et al. 2015) | 1 | 570 mil | CC BY-ND | Desarrollo y evaluación de modelos para comprensión de oraciones. | Flickr30k corpus |
| MEANTIME | (Minard et al. 2016) | 4 | 120 | CC-BY | Generación automática de textos | Portal Wikinews |
| ACR | (He, McAuley, 2016) | Varios. No se especifica. | 130 millones en total | Acceso: abierto. Licencia personalizada: términos de uso de amazon.com | Propósitos académicos y de investigación | Comentarios de usuarios de la plataforma |

| | | | | | | |
|---|---|---|---|---|---|---|
| XTREME | (Hu et. al 2017) | 40 | 15 GB en total | Varias, según el grupo de datos: CC-BY CC BY-NC CC BY-ND | Benchmarks en 9 tareas | Múltiples corpus de gran escala. |
| Multilingual Corpus of Online Educational Content | (Sosoni et al. 2018) | 11 | 87 mil | Personalizada: h2020 Open Research Data Pilot | Tareas de traducción por máquina | Iversity.org Videolectures.net Coursera QED |
| XNLI | (Conneau et al. 2018) | 15 | 10 mil | Varias, según el grupo de datos: CC BY-ND Personalizada: OANC. | Desarrollo y evaluación de modelos multilingües para comprensión de oraciones. | MultiNLI |
| MultiNLI | (Williams, Nangia, Bowman, 2018) | 1 | 433 mil | Varias, según el grupo de datos: CC BY-ND Personalizada: OANC. | Desarrollo y evaluación de modelos para comprensión de oraciones. | Open American National Corpus (OANC) |
| Yelp Open Dataset | (Yelp 2019) | Varios. No se especifica. | 8 millones en total | Apache-2.0 | Propósitos académicos y de investigación | Comentarios de usuarios de la plataforma |
| RCV2 | (Reuters 2019) | 13 | 487 mil | Acceso: privado. Licencia: bajo términos de uso de Reuters News | Desarrollo de investigación en NLP | Artículos de Reuters News |
| MLQA | (Lewis et al. 2019) | 7 | 12 mil en inglés y 5 mil para cada idioma restante | CC BY-NC | Soporte al desarrollo de sistemas de preguntas y respuestas. | Wikipedia |

| | | | | | | |
|---|---|---|---|---|---|---|
| MuST-C | Cattoni et al. 2020) | 14 | 270 mil | CC BY-NC-ND | Tareas de traducción por máquina. | Charlas TED |
| MARC | (Keung et al. 2020) | 6 | 210 mil | Acceso: abierto. Licencia personalizada: términos de uso de amazon.com | Clasificación de acuerdo al número de estrellas. | Amazon Reviews Customer |
| MLSUM | (Scialom et al. 2020) | 5 | 200 mil | Desconocida | Tareas de sumarización automática de textos. | Portales web de noticias |
| XGLUE | (Liang et al. 2020) | 89 | 2 TB en total | Varias, según el grupo de datos: CC-BY CC BY-NC | Benchmarks en 11 tareas | Múltiples corpus de gran escala. |

# Bibliografía

AbuKausar, M., S. Dhaka, V., y Kumar Singh, S. (2013). Web Crawler: A Review. *International Journal of Computer Applications*, 63(2):31–36.

Albán, J., Garcia, D., y Tapia, J. (2020). COSTOS DE IMPORTACIÓN DE PRODUCTOS TEXTILES Y SU INCIDENCIA EN LA UTILIDAD EMPRESARIAL. *Universidad Ciencia y Tecnología*, 24(105):12–19.

Albaum, G. (1997). The Likert Scale Revisited. *Market Research Society. Journal.*, 39(2):1–21.

Albawi, S., Mohammed, T. A., y Al-Zawi, S. (2018). Understanding of a convolutional neural network. In *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, volume 2018-January, pages 1–6. Institute of Electrical and Electronics Engineers Inc.

Amazon (2020). Amazon Customer Reviews Dataset.

Amazon (2021). Open Data on AWS.

Angulo, S. (2021). El sector textil perdió $ 500 millones en 2020.

Aswani, R., Kar, A. K., Ilavarasan, P. V., y Dwivedi, Y. K. (2018). Search engine marketing is not all gold: Insights from Twitter and SEOClerks. *International Journal of Information Management*, 38(1):107–116.

AWS (2021). Registry of Open Data on AWS.

Balahur, A., Fondazione, M. T., y Kessler, B. (2013). Improving Sentiment Analysis in Twitter Using Multilingual Machine Translated Data. Technical report.

Balahur, A. y Turchi, M. (2014). Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Computer Speech and Language*, 28(1):56–75.

Banerjee, S. y Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization.*

Bengfort, B., Bilbreo, R., y Ojeda, T. (2018). *Applied Text Analysis with Python.* O'Reilly, first edition.

Bensouda, Y., Shyu, E., y Kaiser, L. (2019). Traditional Language models | Coursera.

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., y Ojha, A. K. (2020). Developing a Multilingual Annotated Corpus of Misogyny and Aggression. *arXiv.*

Bonaccorso, G., Fandango, A., y Shanmugamani, R. (2018). *Python: Advanced Guide to Artificial Intelligence: Expert machine learning systems and intelligent agents using Python.* Packt Publishing Ltd.

Bowman, S. R., Angeli, G., Potts, C., y Manning, C. D. (2015a). A large annotated corpus for learning natural language inference. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., y Bengio, S. (2015b). Generating Sentences from a Continuous Space. *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, pages 10–21.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., y Amodei, D. (2020a). Language Models are Few-Shot Learners. *arXiv*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Brownlee, J. (2017). What is the Difference Between Test and Validation Datasets?

Brownlee, J. (2019). 14 Different Types of Learning in Machine Learning.

Burkov, A. (2020). *Machine Learning Engineering*. True Positive Incorporated.

Cao, J., Jin, X., Gao, X., Sun, J., y Zhou, J. (2010). Multilingual parallel corpus of UN documents for contrastive and translation studies. In *Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, PIC 2010*, volume 1, pages 208–212.

Cattoni, R., Di Gangi, M. A., Bentivogli, L., Negri, M., y Turchi, M. (2020). MuST-C: A multilingual corpus for end-to-end speech translation. *Computer Speech and Language*, 66:101155.

Chafe, W. (2011). *The importance of corpus linguistics to understanding the nature of language*. De Gruyter Mouton.

Chassagnon, G., Vakalopolou, M., Paragios, N., y Revel, M. P. (2020). Deep learning: definition and perspectives for thoracic imaging.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., y Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., y Cho, K. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Technical report.

Conneau, A., Lample, G., Rinott, R., Schwenk, H., Stoyanov, V., Williams, A., y Bowman, S. R. (2018). XNLI: Evaluating Cross-lingual Sentence Representations. Technical report.

Cunningham, P., Cord, M., y Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia*, pages 21–49. Springer.

Da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., y dos Reis Alves, S. F. (2017). Artificial Neural Network Architectures and Training Processes. In *Artificial Neural Networks*, pages 21–28. Springer International Publishing.

Das, S. (2009). Importance of flammability, care label and fibre content of apparel. In *Quality Characterisation of Apparel*, pages 103–124. Elsevier.

ECMA-404 (2021a). JSON.

ECMA-404 (2021b). JSON Lines.

Eisele, A. y Chen, Y. (2010). MultiUN: A Multilingual Corpus from United Nation Documents. Technical report.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

Facebook (2021). FAQ - Graph API.

Fair, A. F. y Gardent, C. (2020). Multilingual AMR-to-Text Generation. Technical report.

Forbes (2019). Amazon Surpasses Walmart As The World's Largest Retailer.

GitHub (2021). ¿Cuál es mi cuota de disco? - GitHub Docs.

GNU (2021). GNU Lesser General Public License v3.0 - GNU Project - Free Software Foundation.

Gómez, S. (2021). La Industria textil en el Ecuador - Enrique Ortega Burgos.

Google Cloud (2021). Cloud Translation | Google Cloud.

Graën, J. ., Batinić, D. ., y Volk, M. (2014). Cleaning the Europarl Corpus for Linguistic Applications.

Hogan, A. (2020). Web of data. In *The Web of Data*, pages 15–57. Springer.

Hu, B., Lu, Z., Li, H., y Chen, Q. (2014). Convolutional Neural Network Architectures for Matching Natural Language Sentences. *Advances in neural information processing systems*.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., y Xing, E. P. (2017). Toward Controlled Generation of Text. Technical report.

Jin, B., Sahni, S., y Shevat, A. (2018). *Designing Web APIs*. O'Reilly.

Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3(6):714–717.

Keras (2021). Keras: the Python deep learning API.

Keung, P., Lu, Y., Szarvas, G., y Smith, N. A. (2020). The Multilingual Amazon Reviews Corpus. Technical report.

Kilgarriff, A. y Yallop, C. (2001). What's in a Thesaurus Kelly-KEywords for Language Learning for Young and adults alike View project. Technical report.

Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., y Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8):721–734.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. Technical report.

Kotsiantis, S. B., Kanellopoulos, D., y Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International journal of computer science*, 1(2):111–117.

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Lavie, A. (2011). Evaluating the Output of Machine Translation Systems. Technical report.

Lecun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lewis, P., Oğuz, B., Rinott, R., Riedel, S., y Schwenk, H. (2019). MLQA: Evaluating Cross-lingual Extractive Question Answering. *arXiv*.

Li, J., Galley, M., Brockett, C., Gao, J., y Dolan, B. (2015). A Diversity-Promoting Objective Function for Neural Conversation Models. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 110–119.

Li, Y., Pan, Q., Wang, S., Yang, T., y Cambria, E. (2018a). A Generative Model for category text generation. *Information Sciences*, 450:301–315.

Li, Y., Pan, Q., Wang, S., Yang, T., y Cambria, E. (2018b). A generative model for category text generation. *Information Sciences*, 450:301–315.

Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., Fan, X., Zhang, R., Agrawal, R., Cui, E., Wei, S., Bharti, T., Qiao, Y., Chen, J.-H., Wu, W., Liu, S., Yang, F., Campos, D., Majumder, R., y Zhou, M. (2020). XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation. Technical report.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text summarization branches out*.

Lin, C.-Y. y Och, F. J. (2004). Automatic Evaluation of Machine Translation Quality Using Longest Com-mon Subsequence and Skip-Bigram Statistics. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Liu, C. J. y Han, S. (2012). Bilingual corpus research on chinese english machine translation in computer centres of chinese universities. In *Proceedings - 2012 International Conference on Computer Science and Service System, CSSS 2012*, pages 1720–1723.

Logeswaran, L., Honglak, L., y Bengio, S. (2018). Content preserving text generation with attribute controls. Technical report.

Mancosu, M. y Vegetti, F. (2020). What You Can Scrape and What Is Right to Scrape: A Proposal for a Tool to Collect Public Facebook Data. *Social Media + Society*, 6(3):205630512094070.

McAuley, J. (2018). Amazon review data.

McEnery, T. (2012). Corpus Linguistics. In *The Oxford Handbook of Computational Linguistics*, volume 9780199276349. Oxford University Press.

Miangah, T. M. (2009). Constructing a large-scale english-persian parallel corpus. *Meta*, 54(1):181–188.

Microsoft Azure (2021). Translator | Microsoft Azure.

Minard, A.-L., Speranza, M., Urizar, R., na Altuna, B., van Erp, M., Schoen, A., van Son, C., y Bruno Kessler, F. (2016). MEANTIME, the NewsReader Multilingual Event and Time Corpus. Technical report.

Mitchel, R. (2018). *Web Scraping with Python*. O'Reilly, second edition.

MTurk (2021). Amazon Mechanical Turk.

Ng, A., Katanforoosh, K., y Bensouda, Y. (2017). Deep Learning [MOOC].

Nguyen, G., Dlugolinsky, S., Bobák, · . M., Tran, V., Álvaro, · ., García, L., Heredia, I., Malík, · . P., y Hluchý, · . L. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, 52:77–124.

Ni, J., Li, J., y McAuley, J. (2020). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 188–197. Association for Computational Linguistics.

O'Keeffe, A. y McCarthy, M. (2010). *The Routledge Handbook of Corpus Linguistics*.

One Hour Translation (2021). Pricing | One Hour Translation Enterprise.

Open Source Initiative (2021a). Apache License, Version 2.0 | Open Source Initiative.

Open Source Initiative (2021b). The MIT License | Open Source Initiative.

OpenAI (2019). Better Language Models and Their Implications.

Ordóñez, M. (2015). Los dos lados de la tela.

Pak, A. y Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Technical report, Cedex, Orsay.

Papers With Code (2021). PWC Dataset Licensing Guide | Papers With Code.

Papineni, K., Roukos, S., Ward, T., y Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. Technical report.

Prabhumoye, S., Black, A. W., y Salakhutdinov, R. (2020). Exploring controllable text generation techniques. *arXiv preprint arXiv:2005.01822*.

PyTorch (2021). PyTorch.

Ray, S. K., Ahmad, A., y Kumar, C. A. (2019). Review and Implementation of Topic Modeling in Hindi. *Applied Artificial Intelligence*, 33(11):979–1007.

Remy, E. (2019). How public and private Twitter users in the U.S. compare — and why it might matter for your research | by Emma Remy | Pew Research Center: Decoded | Medium.

Reuters (2019). Reuters Corpora @ NIST.

Sayce, D. (2020). The Number of tweets per day in 2020 | David Sayce.

Schroepfer, M. (2018). An Update on Our Plans to Restrict Data Access on Facebook. Technical report.

Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., y Staiano, J. (2020). MLSUM: The Multilingual Summarization Corpus. *arXiv*.

Shafranovich, Y. (2005). Common Format and MIME Type for Comma-Separated Values (CSV) Files.

Sosoni, V., Kermanidis, K. L., Stasimioti, M., Naskos, T., Takoulidou, E., Van Zaanen, M., Castilho, S., Georgakopoulou, P., Kordoni, V., y Egg, M. (2018). Translation Crowdsourcing: Creating a Multilingual Corpus of Online Educational Content. Technical report.

Statista (2016). Ranking de webs donde opinaron más usuarios 2016 | Statista.

Statista (2021). Twitter: most users by country | Statista.

Statistical Machine Translation (2017). Moses - Main/HomePage.

Surcin, S., Hamon, O., Hartley, A., Rajman, M., Popescu-Belis, A., Mustafa, W., Hadi, E., Timimi, I., Dabbadie, M., y Choukri, K. (2005). Evaluation of Machine Translation with Predictive Metrics beyond BLEU/NIST: CESTA Evaluation Campaign 1. *Evaluation of machine translation with predictive metrics beyond BLEU/NIST: CESTA evaluation campaign 1.*

TensorFlow (2021a). Por qué TensorFlow.

TensorFlow (2021b). SparseCategoricalCrossentropy.

Twitter (2021). Información sobre las API de Twitter.

Universidad de Cuenca (2020). Investigación | Universidad de Cuenca.

UPAEP (2021). Cómo trabajar con texto delimitado por tabulaciones (tsv) - UPAEP - Google Apps.

Van Der Smagt, P. y Krose, B. (1996). An introduction to Neural Networks. Technical report.

Vieira, A. y Ribeiro, B. (2018). *Introduction to deep learning business applications for developers.* Springer.

Vogel, S., Zhang, Y., y Waibel, A. (2018). Interpreting BLEU/NIST scores: How much improvement do we need to have a better system Speech Processing View project Parameter Optimization for Statistical Machine Translation View project Interpreting BLEU/NIST Scores: How Much Improvement Do We Need to Have a Better System? Technical report.

W3C (2015). XML Essentials - W3C.

Williams, A., Nangia, N., y Bowman, S. R. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. Technical report.

Yang, Q., Huo, Z., Shen, D., Cheng, Y., Wang, W., Wang, G., y Carin, L. (2019). An end-to-end generative architecture for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3132–3142.

Yelp (2019). Yelp Dataset.

Yelp (2021). Restaurants, Dentists, Bars, Beauty Salons, Doctors - Yelp.

YouTube (2021). YouTube Data API.