Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Desarrollo de modelos para el descubrimiento de relaciones de conteo y de medidas en textos de discursos científicos

Tesis presentada para obtener el título de:
Maestría en Ciencias de la Computación

Presenta:
Cybele Neves Moutinho

Asesora de Tesis:
Dra Darnes Vilariño Ayala

Co-asesora de Tesis:
Dra. María Josefa Somodevilla García

Enero 2023

Agradecimientos

Agradezco al Todo, a la Benemérita Universidad Autónoma de Puebla, al Laboratorio Nacional de Supercómputo del Sureste de México, a todos los Doctores y Maestros que aportaron conocimiento y herramientas para realizar este proyecto.

En especial, agradezco a mi marido que me incentivo y motivo del inicio al fin en esta jornada. Gracias por acompañarme en este logro a más.

Resumen

Las redes neuronales para el modelado del lenguaje han demostrado su eficacia en varias subtareas del procesamiento del lenguaje natural. Sin embargo, el entrenamiento de modelos modelos lingüísticos profundos lleva mucho tiempo y es muy intensivo desde el punto de vista informático. Los modelos lingüísticos preentrenados como BERT, resultan muy atractivos porque ofrecen un rendimiento de vanguardia y liberan a los profesionales de la carga de preparar los recursos adecuados (tiempo, hardware y datos) para entrenar los modelos.

La idea de este trabajo es estudiar las investigaciones recientes que se han centrado en la automatización de la extracción de información de textos en lenguaje natural mediante el Procesamiento del Lenguaje Natural (PLN), en especifico para la tarea de SEMEVAL Task8 del año 2021 la cual requiere de un conjunto especifico de subtareas las cuales giran en torno al la detección de la unidad, la medida y las relaciones semánticas que se pueden aplicar a estas. Este proceso requiere una gran cantidad de conocimiento del dominio. En general, el PLN se emplea para convertir automáticamente la información almacenada en lenguaje natural a un formato comprensible para la máquina. El objetivo principal del PLN es extraer conocimiento de datos no estructurados muy ambiguos con gramáticas complejas. El procesamiento del lenguaje natural es un campo cada vez más importante con aplicaciones crecientes como la búsqueda, la traducción automática y la interacción general entre el ser humano y el ordenador. También es un campo de la informática y la lingüística relacionado con la la Inteligencia Artificial (IA) y la Lingüística Computacional (LC).

Sin embargo, como los modelos de red son genéricos, pueden en ámbitos específicos. En este estudio, investigamos el caso de la clasificación de textos multiclase y sus

relaciones con entidades, una tarea relativamente menos estudiada en la literatura que evalúa modelos lingüísticos previamente modelados.

Esta tarea representa un reto para de los modelos genéricos preentrenados (BERT, RoBERTa, SciBERT) para clasificar una parte del conjunto de datos de documentos, investigamos la intuición de que un modelo especializado preentrenado para documentos científicos produce mejoras en comparación con los modelos genéricos BERT. La cual fue confirmada al utilizar el modelo SciBERT el cual se encuentra entrenado con documentos científicos por lo cual presenta resultados de alrededor de 4 % mejores que los modelos implementados de BERT y RoBERTa.

El modelo presentado fue implementado en su parte de red neuronal con cada una de las modelos anteriormente mencionados (BERT, SciBERT y RoBERTa) teniendo como objetivo lograr identificar el mejor de ellos para el tratamiento de problemas de este estilo, comprobando con base en los resultados obtenidos que el modelo que utiliza SciBERT tiene un comportamiento mejor en comparación con los otros modelos, una explicación del éxito que tuvo esta relacionada a que el sistema SciBERT fue entrenado y cuenta con adaptaciones para el procesamiento de documentos científicos, por lo tanto tiene una ligera ventaja respetos a los demás. El uso de herramientas de desarrollo como Python y Frameworks de aprendizaje profundo como pytorch y pytorch-crf permiten el desarrollo ágil de modelos de procesamiento de textos de forma eficiente y rápida, el factor en contra es que requieren de arquitecturas de hardware de gran desempeño, en especifico aquellas que cuentan con aceleradores GPU, pero gracias a la existencia de Laboratorios de supercómputo como el LNS de la BUAP se logro el entrenamiento y ejecución de los modelos propuestos en este trabajo de forma satisfactoria.

Mayor trabajo, investigación y recursos son necesarios para poder obtener modelos que completen este tipo de tareas con un nivel satisfactorio, la mayor calificación actual para este tipo de tareas de forma general solo supera el 0.5 de éxito, lo que representa muy poco para un sistema que deseablemente seria automático. Al día de hoy el área de PLN es de amplia investigación y con los esfuerzos necesarios así como el avance en las tecnologías de hardware harán posibles, en un tiempo relativamente corto, modelos mucho mas complejos y robustos que permitan la obtención de resultados favorables.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El área de Procesamiento del Lenguaje Natural (PLN) implica la creación de sistemas informáticos para realizar tareas significativas con el lenguaje natural y comprensible para los humanos. Se utiliza en campos importantes como el procesamiento de texto y la lingüística computacional, centrados en la resolución de problemas que involucran la semántica del texto a analizar.

En esta investigación, se propone la identificación y extracción de cantidades, medidas, unidades, atributos de las cantidades y propiedades en los corpus de textos planos para desarrollo de modelos que descubren relaciones de conteo y de medidas en textos de discursos científicos.

1.1. Planteamiento del problema

Una problemática que se presenta en el PLN es el reconocimiento y extracción de información centrada, como encontrar cantidades, medidas, atributos de las cantidades, propiedades de las mismas y el contexto en que son utilizadas. El descubrimiento de información en textos no estructurados es un problema latente en este siglo, ya que mucha información que se maneja en la actualidad carece de alguna estructura.

A nivel internacional se están realizando muchos esfuerzos buscando construir aplicaciones inteligentes que sean capaces de "entender" el sentido de un texto en su totalidad.

En este trabajo, se investigan técnicas de extracción de información para desa-

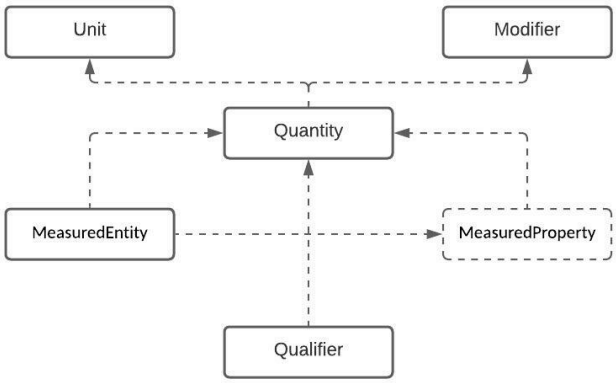

Figura 1.1: Modelo de anotación

rrollar modelos que permitan el descubrimiento de relaciones de conteo y de medidas en textos de discursos científicos en el idioma inglés.Se busca descubrir anotaciones entre cantidades y sus relaciones. Para resolver el problema se debe crear un modelo de anotación, como se observa en el cuadro 1.1, que consta en etiquetar Cantidad (*Quantity (Q)*), Entidad Medida (*MeasuredEntity (ME)*), Propiedad Medida (*MeasuredProperty (MP)*), opcionalmente un Calificador (*Qualifier (Qr)*) y las relaciones *HasEntity, HasProperty, Qualifier*.

Los valores también pueden tener atributos adicionales de *Modifier* (modificadores de la cantidad) como *isMean, isApproximate, isCount,isRange, isList, isMedian*, etc. . .

Cuadro 1.1: Alcance entre cantidades y relaciones entre ellas

Alcance entre cantidades
Cantidad (Quantity Q)
Medida de entidad (MeasuredEntity ME)
Propiedad de la medida (MeasuredProperty MP)
Calificador (Qualifier Qr)
Unidad (Unit)

Relaciones entre ellas
TieneCantidad(HasQuantity HQ)
TienePropiedad(HasProperty HP)
Califica a (Qualifies Qs)

Las relaciones, como se observan en la Figura 1.1, donde *Quantity* puede estar directamente relacionada con una *MeasuredEntity*, o pueden estar indirectamente

relacionadas con una *MeasuredEntity* a través de una *MeasuredProperty*. *Qualifier* proporciona información adicional necesaria para interpretar la medición. Estos incluyen detalles como la presión a la que se observó un punto de ebullición, o la profundidad y el lugar donde se tomó una muestra del océano. Dado que los textos, pueden contener diferentes partes de esta información, todas las relaciones son opcionales. Una *MeasuredEntity* se puede relacionar con una *MeasuredProperty* o una *Quantity*, una *MeasuredProperty* se puede relacionar con una *Quantity*, y *Qualifier* puede relacionarse con cualquier *MeasuredEntity, MeasuredProperty* o *Quantity* mediante una relación de califica *Qualifies*.

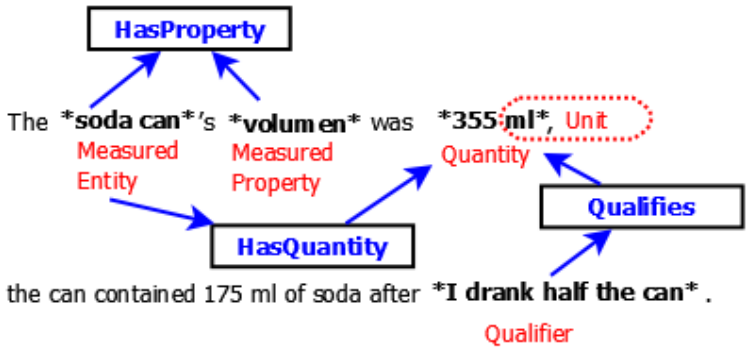

Figura 1.2: Frase con ejemplo de anotación.

Un ejemplo de anotación se puede observar en la en la Figura 1.2, con la siguiente frase en inglés *"The soda can's volumen was 355 ml, the can contained 175 ml of soda after I drank half the can."*, en rojo están las anotaciones y en los cuadros azules están las relaciones entre ellas, conforme descrito arriba. Una Cantidad (Q) puede ser un recuento o una medida, y las medidas se componen de una unidad y un valor. Para Q se encuentra ***355 ml*** con su *Unit* ***ml***, así como una medida de entidad, *ME* ***soda can***, su medida de propiedad *MP* ***volumen***, y la segunda parte de la frase hay *Qr*, que modifica la cantidad inicial con ***I drank half the can***.

En las relaciones esta la entidad de medida *ME* ***soda can*** que tiene relación de cantidad *HP* con ***volumen***, también una relación de propiedad *HQ* con ***355 ml***, y el calificador *Qr* **I drank half the can** califica *Qs* la *Q* ***355 ml***.

1.2. Objetivos

Para alcanzar los resultados deseados se plantean el objetivo general que define el alcance de la investigación y los objetivos específicos que detallan los procesos esenciales para la completa realización del trabajo.

1.2.1. Objetivo General

Desarrollar modelos que permitan descubrir cantidades, medidas, unidades, atributos de las cantidades y propiedades de las mismas, en diferentes dominios de texto, para que se logren entender las relaciones semánticas entre cantidades que se presentan en textos planos.

1.2.2. Objetivos Específicos

1. Analizar técnicas para la identificación y extracción de cantidades, medidas, unidades, atributos de las cantidades y propiedades en los corpus de textos planos.

2. Implementar un método para la extracción de términos de cantidades, medidas, unidades, atributos de las cantidades y propiedades en los corpus de textos planos.

3. Implementar un método para la extracción de relaciones de los términos cantidades, medidas, unidades entre los términos de atributos de las cantidades y propiedades.

4. Validar el comportamiento de los modelos en los diferentes dominios de los textos planos, para validar la relación que existe entre los conceptos y la forma de escritura dentro de cada tipo de dato.

1.3. Antecedentes

Los recuentos y las mediciones son una parte importante del discurso científico, aunque sea relativamente fácil encontrar medidas en el texto, una medida simple

por si sola no es informativa sin saber a qué se refiere, además, la ubicación de esta información relativa a la medición puede variar mucho e incluso puede estar en una oración diferente. (Harper, 2021)[18].

SemEval es una serie de talleres internacionales de investigación de PLN con objetivo avanzar en el estado actual del arte en el análisis semántico y ayudar a crear conjuntos de datos anotados de alta calidad para resolver problemas en la semántica del lenguaje natural. El taller de cada año presenta una colección de tareas compartidas en las que se presentan y comparan sistemas de análisis semántico computacional diseñados por diferentes equipos.

Recientemente, en SemEval 2021 presentaron la Tarea 8 MeasEval: Recuentos y Medidas, que se compone de cinco subtareas que cubren la extracción de tramos, la clasificación y la extracción de relaciones, dado un párrafo de un texto científico:

1. Para cada párrafo de texto, identificar todos los intervalos de cantidad.

2. Para cada Cantidad identificada, identificar la Unidad de medida, si existe y clasificar los Modificadores de valores adicionales (recuento, rango, aproximado, promedio, etc.) que se aplican a la Cantidad.

3. Para cada Cantidad identificada, identificar la Entidad medida a la que se aplica (si existe) y marcar su intervalo. Si también existe una propiedad medida asociada, identificándola y marcando su extensión también. Caso exista la Entidad medida y/o Propiedad medida marcar su intervalo.

4. Identificar y marcar el intervalo de cualquier Calificador que sea necesario para registrar un contexto relacionado adicional para validar o comprender cada Cantidad identificada.

5. Identifique las relaciones entre los intervalos de cantidad, Entidad medida *(MeasureEntity)*, Propiedad medida *(MeasuredProperty)* y Calificador *(Qualifier)* utilizando los tipos de relación HasQuantity, HasProperty y Qualifies.

Esta investigación se relaciona en partes con las subtareas propuestas, ya que se pretende identificar las cantidades, medidas, unidades, atributos de las cantidades y propiedades de las mismas, en diferentes dominios de texto, para que se logren

entender las relaciones semánticas entre cantidades que se presentan en los textos planos.

1.4. Justificación

El PLN es un tema muy discutido e investigado en la actualidad. Como es una de las áreas de investigación más antiguas en aprendizaje automático, se utiliza en campos importantes como el procesamiento de texto y la lingüística computacional. El PLN trajo un gran avance en el campo de la computación y la Inteligencia Artificial (IA) y varios algoritmos utilizados para el PLN dependen principalmente de la red neuronal, a máquina aprende la sintaxis y el significado del lenguaje humano, lo procesa y le da la salida al usuario. El área de PLN implica la creación de sistemas informáticos para realizar tareas significativas con el lenguaje natural y comprensible para los humanos (Jain et al., 2018)[2].

En el área de base de datos y recuperación de la información se distingue una línea de investigación que lleva varios años, centrada en la resolución de problemas que involucran la semántica del texto a analizar.

Se han desarrollado diferentes modelos para el planteamiento de aplicaciones inteligentes que permitan resolver el problema de similitud semántica, implicación textual, búsqueda de predadores en la red y el manejo de problemas de procesamientos de lenguaje natural entre idiomas y en diferentes idiomas.

Capítulo 2

Estado del Arte

El estado del arte tiene como objetivo conocer investigaciones anteriores que desarrollaron proyectos extracción de medidas, atributos y sus relaciones, para proveer un punto de partida en esta investigación y ayudar a la creación de enfoques para el descubrimiento de relaciones de conteo y de medidas en textos de discursos científicos.

Recientemente se han hecho intentos para identificar la entidad nombrada y la propiedad que se está midiendo; Hundman y Mattmann, en 2017,[21] publicaron un sistema de extracción de valores de medición, unidades y palabras relacionadas del texto en lenguaje natural. Utilizan campos aleatorios condicionales, *Conditional Random Fields (CRF)*, para identificar valores de medición y unidades, seguido de un sistema basado en reglas para encontrar entidades relacionadas, descriptores y modificadores dentro de una oración. Aunque demuestran una buena capacidad para generar extracciones de precisión con una fuerte recuperación, pretenden mejorar de los requisitos de medición, para la misión propuesta por la NASA. Su sistema está relacionado con herramientas Grobid, una biblioteca que utiliza campos aleatorios condicionales lineales para identificar unidades de medida y valores, y CoreNLP, una paquetería de Stanford NLP para anotaciones lingüísticas de textos en Java. En general generan, errores ya que es necesario, un etiquetado manual y uso de conocimiento de expertos lingüistas.

Por otra parte, Augenstein et al (2017) [4],en su publicación ofrece una solución etiquetando y clasificando frases clave con sus las relaciones; para ello presentan 26

sistemas en 3 escenarios de evaluación. La mayoría de ellos utilizan red neuronal (RNN), en combinación con *Convolutional NN (CNN)*, *Support Vector Machines (SVM)* y CRF, con un conjunto de características léxicas bien diseñadas. La investigación realizada concluye en que los éxitos de los sistemas varían en sus enfoques; en este trabajo, el encontrar las frases claves es una tarea muy compleja, dado que el conjunto de datos contiene muchas frases clave largas y poco frecuentes, y los sistemas que se basan en recordarlas no funcionan bien.

Hogan et al., (2021) [19], publicó su estudio de investigación Grafos de Conocimiento *(Knowledge Graphs)*, el cual describe la creciente aceptación tanto de la industria como de la academia, en escenarios que requieren la explotación de diversos tipos de datos, con la idea central de usar grafos para representar datos, a menudo mejorado con alguna forma de representar explícitamente el conocimiento, mediante una combinación de técnicas deductivas e inductivas. La investigación en extracción de información y conocimiento, la creación de grafos se ha concentrado en formar tripletas extrayendo entidades y relaciones (Konstantinova, 2014)[24]. Su estudio concluye que los desafíos para los grafos de conocimiento incluyen escalabilidad, particularmente para el razonamiento deductivo e inductivo; calidad, no tanto en términos de datos, sino también modelos de grafos de conocimiento; diversidad, como la gestión contextual; y dinamización, considerando datos en tiempo real.

En forma general para el *SemEval 2021 Task 8 MeasEval*: Recuentos y Medidas se presentaron 25 proyectos y 9 de ellos, se incluyeron en el artículo de discusión de la metodología y modelo propuesto para resolver el problema presentado en esta tarea 8.

Todos los trabajos presentados obtuvieron buenos resultados en la obtención de Cantidades y Unidades, aunque los elementos contextuales *MeasuredEntity, MeasuredProperty, Qualifier*, y sus relaciones, son un desafío aún mayor; por lo que es necesario crear nuevos modelos para extraer las entidades y sus relaciones; a continuación, se discuten como los diferentes equipos brindaron la solución al problema:

- Cao y su equipo, propusieron una herramienta de extracción de medidas y recuento en cascada, llamada CONNER, que realiza una extracción de cantidad inicial con un conjunto de PointerNet, una arquitectura neuronal para aprender la probabilidad condicional de una secuencia de salida, junto con

CRF. Usan un clasificador basado en BERT, siglas en ingles de *Bidirectional Encoder Representations from Transformers* representaciones de codificador bidireccional de transformadores, diseñado para pre-entrenamiento de representaciones bidireccionales profundas a partir de texto sin etiquetar mediante el condicionamiento en el conjunto del contexto izquierdo y derecho en todas las capas (Devlin, 2019[14]), para el etiquetado de Modifier y un sistema basado en reglas para Unit, seguido de etiquetadores de relaciones específicas con la misma arquitectura que el etiquetador de Quantity (Cao et al., 2021)[8].

- Davletov y su equipo, propusieron un sistema de aprendizaje multitarea en forma de preguntas y respuestas, utilizando aprendizaje por escala para ponderar la contribución de las subtareas, con enfoques basados en modelos LUKE, formado por sus siglas en inglés *Language Understanding with Knowledge-based Embeddings*, modelo de representación contextualizada pre-entrenada de palabras y entidades basada en transformador, para extraer intervalos de cantidad; y RoBERTa, siglas en inglés de *Robustly Optimized BERT Pretraining Approach* (Liu, 2019)[29], que extrae todo lo relacionado con las cantidades encontradas e incluidas las cantidades mismas (Davletov et al., 2021)[11] .

- Gangwar y su equipo, de manera similar, utilizan un etiquetador de secuencia SciBERT, basado en BERT previamente capacitado para realizar tareas científicas y CRF para las cantidades. Para los modificadores utiliza SciBERT con BiLSTM, por sus siglas en ingles *Bidireccional Long Short-Term Memory* memoria larga a corto plazo bidireccional, basado en caracteres para etiquetado de unidades (Gangwar et al., 2021)[16].

- Therien y equipo, usan un sistema de canalización de diferentes módulos de aprendizaje automático y basados en reglas, con SciBERT en un clasificador de clases múltiples, a nivel de token, en todas las clases de los intervalos, lo que da inferencias conjuntas entre los distintos tipos de intervalos. (Therien et al., 2021)[37].

- El equipo de Avram, propone un sistema en cascada compuesto por subsistemas individuales, utiliza RoBERTa junto con CRF para extracción de la cantidad. Usan un BiLSTM para extraer unidades y modificadores de clasificadores, y

luego usan un sistema de respuesta a preguntas con plantilla, como una entidad conjunta con un sistema de extracción de relaciones. (Avram et al., 2021)[5].

- Karia y su equipo, usa BioBERT, siglas en inglés *Bidirectional Encoder Representations from Transformers for Biomedical Text Mining*, en general no les funcionó muy bien un clasificador binario en lugar de etiquetas BIO, esquemas de etiquetado de sequencia (*inside, outside, beginning*), es un formato de etiquetado común para etiquetar tokens en una tarea de fragmentación, *chunk*, en lingüística computacional; y capas CRF para el etiquetado de secuencia de cantidad. Los modificadores y las unidades se manejan usando palabras clave y coincidencia de diccionario, mientras que usan un modelo BERT de tareas múltiples para los componentes restantes (Karia et al., 2021)[22].

- Liu y su equipo, participaron solo con las subtareas de encontrar Cantidad, Unidad y Modificador. Usan BERT para el etiquetado de secuencia y GLUE, RACE y SQuAD[1], para Cantidades, usan un esquema de etiquetado de secuencia de entrada y salida, similar en Cantidades para etiquetar Unidades y un clasificador de clases múltiples para clasificar Cantidades a los Modificadores apropiados (Liu, 2021)[28].

- Lathiff y su equipo, pre-procesaron el texto usando GATE, siglas en inglés de *General Architecture for Text Engineering*, son componentes para el procesamiento del lenguaje, como herramientas para visualizar y manipular texto; y ANNIE, siglas de *A General Architecture for Text Engineering, una colección de algoritmos listos para usar que realiza IE en texto no estructurado. IE es un proceso que toma como entrada textos invisibles y produce formato fijo, datos sin ambigüedad como salida (Niat, 2020)[31].*, con reglas personalizadas para limpiar y tokenizar. Trataron los árboles de análisis de dependencias de núcleo de *Stanford Core Dependency Parse trees*, como grafos para extraer subgrafos, comenzando cada consulta de ruta de los tokens para identificar MeasureEntities, MeasuredProperties y Qualifiers, con el uso de Graph CNN, *Graph Convolutional Neural Networks* . Utilizaron en los modelos de CLaCBP, sistema

[1]GLUE, RACE y SQuAD.: colección de recursos, puntos de referencias, para entrenar, evaluar y analizar los sistemas de comprensión del lenguaje natural, inclusive ayuda a responder las preguntas en párrafos u oraciones no estructurados (Lawton, 2020)[27]

de tubería modular del *Computational Linguistics at Concordia* (CLAC) Canada (Therien, 2021)[37], para mapear desde sus tokens hasta los intervalos de anotación para cada tipo, al ensamblar su envío final (Lathiff, 2021)[26].

- Pouran con su equipo, formularon su propia tarea basándose en los datos de MeasEval (Harper, 2021)[18], presentaron la descripción de un sistema que usan un enfoque novedoso, la extracción de relación de medición, *Measurement Relation Extraction* (MRE), que emplea una arquitectura profunda basada en la traducción para inducir dinámicamente las palabras importantes en el documento y clasificar la relación entre un par de entidades, cuyo objetivo es reconocer la relación entre entidades medidas, cantidades y condiciones mencionadas en un documento; presentan una nueva técnica de regularización, basada en *Information Bottleneck* (IB) para filtrar la información ruidosa del conjunto inducido de palabras importantes (Pouran et al. 2021)[33].

A pesar de que otros equipos participaron, sus resultados obtenidos quedaron por debajo del *baseline*[2], sus modelos propuestos no fueron considerados como aportaciones importantes en la solución de este problema. Sigue tabla comparativa de los principales equipos **??**.

[2]Línea de base del SemEval 2021 Tarea 8 (Harper, 2021)[18]

Cuadro 2.1: Tabla comparativa de los principales equipos participantes SemEval 2021 Tarea 8 (Harper, 2021)[18]

Equipo	***Sistema***	***Herramienta***	***Uso***
Davletov y su equipo [11]	Aprendizaje multitarea en forma de preguntas y respuestas, y aprendizaje por escala	LUKE	Extrae cantidades
		RoBERTa	Extrae las relaciones con las cantidades
Cao y su equipo [8]	Extracción de medidas y recuento en cascada	PointerNet y CRF	Extrae cantidades y unidades
		BERT	Extrae modificadores
		Basado en reglas	Extrae relaciones
Gangwar y su equipo [16]	Extracción de medidas y recuento en cascada	SciBERT y CRF	Extrae cantidades
		SciBERT	Extrae modificadores
		LSTM bidireccional	Etiqueta unidades
Therien y su equipo [37]	Canalización de diferentes módulos de aprendizaje automático y basados en reglas.	SciBERT	Para todos los intervalos.
Avram y su equipo [5]	Sistema en cascada compuesto por subsistemas individuales	RoBERTa y CRF	Extrae cantidades
		BiLSTM	Extraer unidades y modificadores
		Plantillas	Extraer relaciones

Continúa en la siguiente página

Equipo	***Sistema***	***Herramienta***	***Uso***
Karia y su equipo [22]	Sistema en cascada y el aprendizaje multitarea	BERT	Extrae cantidades, unidades y relaciones
		Diccionarios	Extraer unidades y modificadores
Liuy su equipo [28]	Sistema de modelos BERT previamente entrenados	BERT	Etiqueta secuencia
		GLUE, RACE y SQuAD	Extrae cantidades
Lathiff y su equipo [26]	Árboles de dependencia en una red de convolución de gráficos profundos para la clasificación de tareas múltiples	GATE y ANNIE	pre-procesamiento del texto
		Análisis del núcleo del árbol de dependencia de Stanford	De grafos para extraer subgrafos
		Graph CNN	Identificar MeasureEntities, MeasuredProperties y Qualifiers
		modelos de CLaCBP	Extraer anotación para cada tipo
Pouran y equipo [33]	Arquitectura profunda basada en la traducción para inducir dinámicamente las palabras importantes en el documento.	Information Bottleneck (IB)	Filtrar la información ruidosa del conjunto inducido de palabras

Capítulo 3

Marco teórico

En este capítulo se presentan de manera general las herramientas y conceptos teóricos estudiados para la realización del presente trabajo de tesis.

3.1. Procesamiento de Lenguaje Natural

El lenguaje es el medio que los humanos se comunican y expresan su raciocinio por medio de la asociación de signos con ciertos significados, usando herramientas como la escritura, las señales y la voz para establecer comunicación. Con esto de determina dos tipos de lenguajes (Beltrán et al., 2021)[7]:

- Lenguaje natural: una forma de comunicación entre seres humanos, una base utilizada para comprender el lenguaje natural son los idiomas como español, portugués, ingles, entre otros, utilizados para relacionarse a través de alguna forma de comunicación sea escrita, oral o no verbal, estos lenguajes están en constante crecimiento sin tener en cuenta las reglas que los suceden.
- Lenguaje formal: lenguaje ajustada estrictamente a reglas establecidas cómo por ejemplo la programación, la lógica, matemáticas, y otros.

El lenguaje natural tiene amplio vocabulario y diversas construcciones gramaticales, con características como la flexibilidad y ambigüedad que permite diversas interpretaciones dependiendo de la situación, que son muy eficientes en la comunicación humana, sin embargo para el procesamiento computacional estas características

dificultan la aplicación de procesos de razonamiento, caracterización y formalización (Gil Leiva et al., 1996) [17].

Según Sosa, el estudio del lenguaje natural se estructura principalmente en 4 niveles de análisis: morfológico, sintáctico, semántico y pragmático. (Sosa, 1997)[35].

- Morfológico: enfocada en estudiar la estructura interna de las palabras, clase y segmentación, como puede ser el reconocimiento de sufijos o prefijos.
- Sintáctico: el estudio de las relaciones estructurales entre palabras. Se etiquetan todos los componentes sintácticos que aparecen en la oración y analiza cómo las palabras se relacionan para formar construcciones gramaticalmente correctas.
- Semántico: estudia el significado del lenguaje, busca establecer la relación que existe entre las formas lingüísticas y el sentido con el que están siendo utilizadas, a una secuencia textual determinada.
- Pragmático: estudia como el contexto influye en la interpretación del significado, entre el lenguaje desde los emisores y receptores.

Sosa complementa que se pueden incluir otros niveles de conocimiento como es la información fonética y fonología, es el estudio de los sonidos según la pronuncia de las palabras, el análisis del discurso, estudia la información y sus interpretaciones del discurso; y, por fin, el conocimiento del mundo, referente interacción comunicativa (Sosa, 1997)[35].

El procesamiento del lenguaje natural es un área de investigación en informática e inteligencia artificial (IA) relacionada con el procesamiento de lenguajes naturales, que implica en traducir el lenguaje natural en datos que una computadora pueda utilizar para aprender sobre el mundo (Lane et al., 2021)[25].El objetivo del PLN es que las máquinas comprendan los textos no estructurados y extraigan la información relevante de esos textos por medio de programas que analicen, entiendan y puedan generar lenguajes que las personas usan de manera habitual.(IIC, 2022)[12].

Según Gil Leiva y Beltrán algunas aplicaciones practicas son el uso de correctores ortográficos automáticos, la traducción automática, sistemas de análisis y recuperación de información (Leiva et al., 1996)[17], generación de nuevo texto, preguntas y respuestas, generar resumen, chatbots entre otros (Beltrán et al., 2021) [7].

Este estudio se enfoca en el procesamiento del análisis semántico del PLN con las etapas y herramientas del procesamiento de minería de texto.

3.2. Minería de Texto

La minería de texto, *Text Mining*, es una herramienta que se origina del procesamiento automático de textos, que localiza y extrae la información más importante y esencial de los documentos, identificando patrones en el texto, como tendencias en el uso de palabras, estructura sintáctica, o descubriendo información que no está explícita dentro del texto y que anteriormente no se conocía (Castillo et al., 2007)[9], con área multidisciplinaria basada en la recuperación de información, aprendizaje automático, estadísticas y la lingüística computacional, que puede ser aplicada a recursos de textos en páginas web, libros, correos electrónicos, reseñas de clientes, artículos, entre otros. (Eito et al.,2004) [15]. Eito y Sendo en su articulo, comenta que la minería de texto es una actividad que complementa a minería de datos, porque muchos fabricantes comerciales toman minería texto como clave para facilitar la comprensión de las necesidades de los clientes, como gestión de reclamaciones, transcripciones de las llamadas registradas en los call-center, patentes, noticias de prensa, etc ...

La minería de datos obtiene información a partir de los patrones y tendencias a partir de grandes volúmenes de información estructurada, como bases de datos relacionales. Sin embargo, la minería de texto pretende buscar estos patrones en corpus textuales o en información no estructurada. Ambas buscan deducir nueva información a partir de la información ya existente, con datos estructurados para minería de datos, e con información no estructurada, textos, para minería de texto. (Eito et al., 2004) [15].

La minería de texto facilita la comprensión de las necesidades de los clientes en la industria comercial y permitir el seguimiento de la competencia con gestión de reclamaciones, transcripciones de las llamadas registradas en los call-center, patentes, noticias de prensa, etc [15]. Centrando la minería de texto con en la ciencia y la investigación, hay más de 50 millones de revistas científicas, inmensa información virtual en páginas web, informes de organizaciones públicas, libros, entre otros, sien-

do cada vez es más difícil a los investigadores hacer un seguimiento de lo que se publica en su propio campo. La minería de textos ayuda a resolver este problema y a encontrar nueva información (Universo Abierto, 2018) [10].

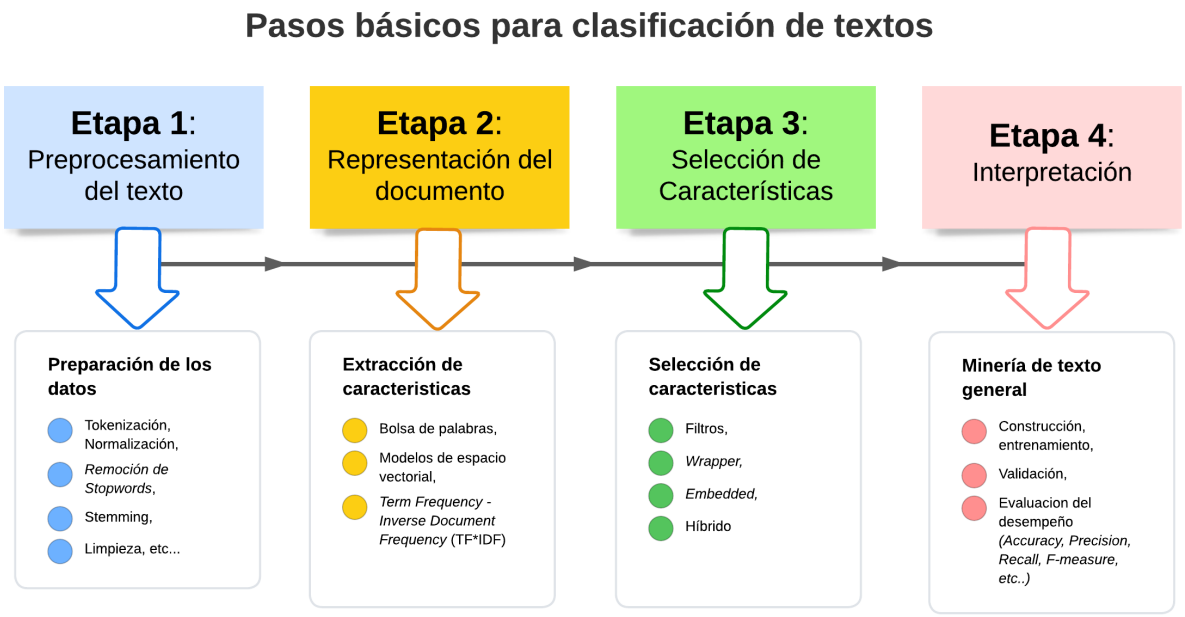

Figura 3.1: Proceso de minería de texto: Pasos básicos para clasificación de textos. (Abiodun, 2021)[1]

La mineria de texto funciona con la extracción de textos puede dividirse en cuatro pasos, n la etapa de implementación se crea el modelo propuesto con proceso de minería de texto y sus pasos para clasificación de textos, como se observa en la Figura 3.1.

1. Etapa 1: Preprocesamiento de datos del texto con la preparación de datos, es la transformación del texto de lenguaje humano a un formato comprensible a máquina para su posterior procesamiento con limpieza, normalización, tokenización, entre otros.

   a) Limpieza: se elimina contenido no deseado o innecesario.

   b) Normalización: transforma diferentes formas a una sola.

c) Tokenización: hace la separación del texto en tókenes (unidades mínimas, por ejemplo palabras), según los espacios en blanco presentes y las puntuaciones.

d) *Stemming*: separa los prefijos y los sufijos de las palabras para derivar su raíz y su significado.

e) Remoción de *Stop Words*: remueve las palabras más comunes del idioma (artículos, preposiciones, ...) que aparecen repetidas veces y no aportan información valiosa.

f) Separación en conjuntos de datos: entrenamiento, validación, prueba.

g) Generación del vocabulario, la lista de tókenes conocidos.

2. Etapa 2: Representación del documento, con extracción de características que implica en extraer información significativa de datos de texto no estructurados y presentarla en un formato estructurado, por medio de modelos, como:

a) Bolsa de palabras (*Bag-of-Words*): Es una representación de texto que describe la ocurrencia de palabras dentro de un documento. Utiliza un vocabulario de palabras conocidas y una medida de la presencia de palabras conocidas.

b) Modelo de espacio vectorial *(Vector Space Model - VSM)*:Es un modelo algebraico para representar documentos de texto como vectores en n-dimensiones, donde cada dimensión corresponde a un término, basado en cálculo de similitud.

c) Frecuencia de término-Frecuencia de documento inversa *TF-IDF (Term Frequency-Inverse Document Frequency)* :es una estadística numérica simple que se utiliza para determinar la relevancia de un texto en relación con los términos en una consulta de búsqueda.

d) Partes del discurso *(Parts of Speech - POS)*: Es el proceso de marcar una palabra en un texto (corpus) como correspondiente a una parte particular del discurso, basado en su definición y su contexto. En la gramática tradicional, POS es una categoría de palabras que tienen propiedades gramaticales similares, explica cómo se usa una palabra en una oración.

3. Etapa 3: Selección de Características, el cual intenta encontrar un subconjunto óptimo de características apropiadas de la amplia gama inicial de características, los tres métodos principales de selección de características para la clasificación de texto: se basan en filtros, envoltorios *(wrapper)* e integrados *(embedded)*.

    a) Filtros: reduce el numero de características independientemente del modelo de clasificación.

    b) Envoltorios *(wrapper)*:envuelven la selección de características alrededor del modelo de clasificación y usan la precisión de predicción *(prediction accuracy)*del modelo para seleccionar o eliminar iterativamente un conjunto de características.

    c) Integrados/Incrustación *(embedded)*: el proceso de selección de características es una parte integral del modelo de clasificación.

4. Etapa 4: Interpretación del modelo,

    a) Entrenamiento: es el material a través del cual la computadora aprende a procesar la información.

    b) Validación: es para validar el modelo construido.

    c) Evaluación del desempeño: Ayuda a encontrar el mejor modelo que representa los datos y qué tan bien funcionará el modelo elegido en el futuro *(Accuracy, Precision, Recall, F-measure, etc..)*

3.3. Transformer

Las redes neuronales recurrentes (RNN) y la memoria a corto plazo, *Long-Short Term Memory (LSTM)*, se establecen con enfoques pioneros en el modelado de secuencias y problemas de transducción como el modelado del lenguaje y la traducción automática, se ha hecho mucha investigación para seguir expandiendo el uso de los modelos de lenguaje recurrentes y las arquitecturas de codificador-descodificador (Luong,2014) [30].

Los mecanismos de atención se han convertido en una parte integral de los modelos de transducción y modelado de secuencias convincentes en varias tareas, lo que permite modelar dependencias independientemente de su distancia en las secuencias de entrada o salida. Sin embargo, con algunas excepciones, estos mecanismos de atención se utilizan junto con una red recurrente.

Los modelos recurrentes generalmente utilizan el cálculo a lo largo de las posiciones de los símbolos de las secuencias de entrada y salida (Ankit,2022) [3]. Esta naturaleza secuencial impide la paralelización dentro de los ejemplos de entrenamiento, lo que se vuelve crítico en longitudes de secuencia más largas, ya que las restricciones de memoria limitan el procesamiento por lotes entre ejemplos (Vaswani et al., 2017) [39].

El Transformer es el primer modelo de transducción basado completamente en la auto-atención para calcular representaciones de su entrada y salida con arquitectura de codificador-decodificador, sin utilizar capas recurrentes, esta especialmente adecuado para la comprensión del lenguaje (Uszkoreit,2017) [38].

Los mecanismos de atención se han vuelto en una parte integral del modelado de secuencias y los modelos de transducción en varias tareas, lo que permite el modelado de dependencias sin tener en cuenta su distancia en las secuencias de entrada o salida, aunque estos mecanismos de atención en algunos casos de utilizan junto con una red recurrente.

En el Transformer, tiene una arquitectura con modelo que evita la recurrencia y, en cambio, se basa completamente en un mecanismo de atención para dibujar dependencias globales entre la entrada y la salida. El Transformer permite una paralelización significativamente mayor y puede alcanzar un nuevo estado del arte en calidad de traducción después de ser entrenado durante tan solo doce horas en ocho GPU P100 (Vaswani et al., 2017) [39]. La autoatención es un mecanismo de atención que relaciona diferentes posiciones de la misma secuencia para calcular una representación de la misma, lo que reduce a un número constante de operaciones.

Vaswani et al., explica en su articulo [39], que la mayoría de los modelos tienen una arquitectura de codificador-decodificador, donde el codificador asigna una secuencia de entrada de representaciones simbólica a una secuencia de representaciones continuas. Con esta secuencia final, el decodificador genera una secuencia de

salida de símbolos, un elemento cada vez. En cada paso el modelo es autorregresivo, es decir, consume los símbolos generados anteriormente como entrada adicional al generar el siguiente paso. El Transformer sigue esta arquitectura, con el uso de capas apiladas de auto-atención y punto a punto, totalmente conectadas, tanto para el codificador como para el decodificador, como se observa en la Figura 3.2.

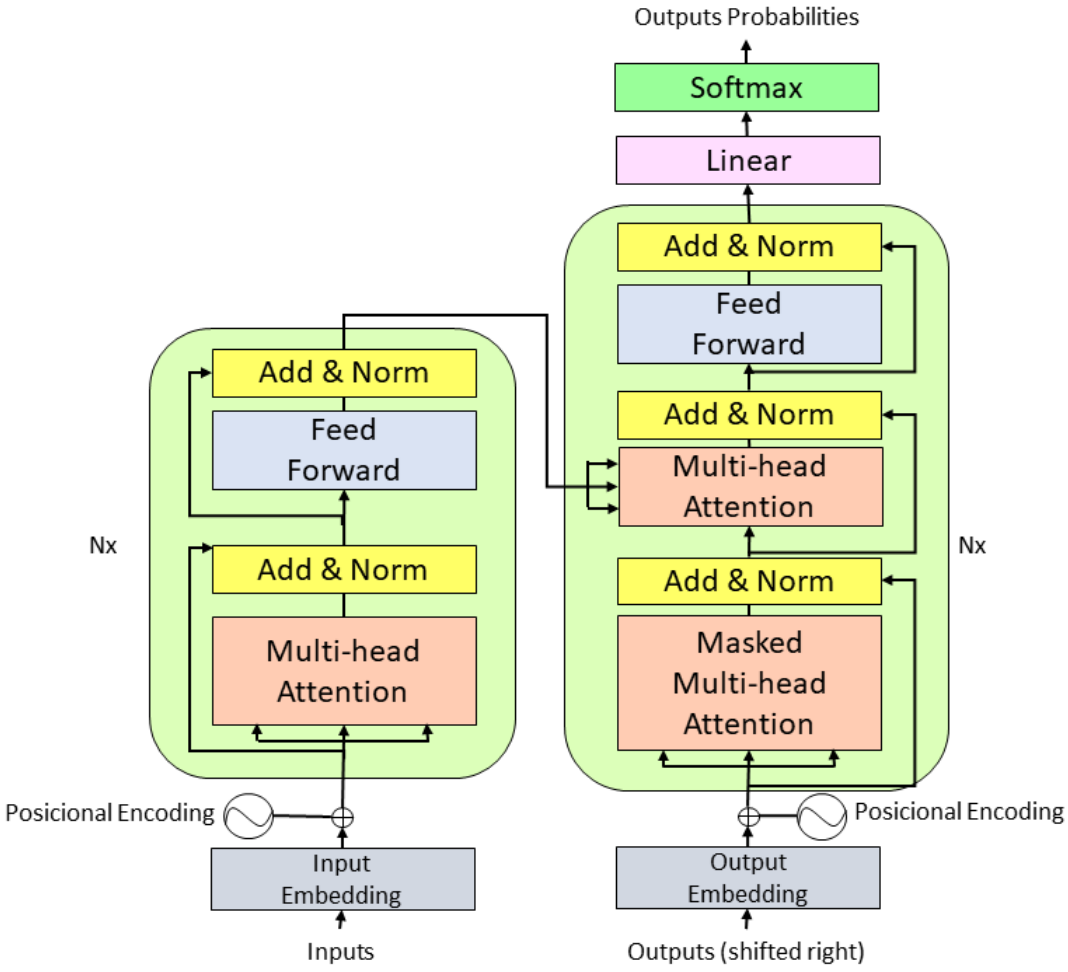

Figura 3.2: Arquitectura de la red Transformer (Vaswani et al., 2017) [39].

El autor sigue explicando que el codificador está compuesto por una pila de N=6 capas idénticas con dos subcapas cada una. El primero es un mecanismo de auto-atención de varios cabezales, y el segundo es una red de realimentación simple, totalmente conectada por posición. Se usa una unión residual alrededor de cada

una de las dos subcapas, seguida de la normalización de las capas. Para facilitar estas conexiones residuales, todas las subcapas del modelo, así como las capas de incrustadas *(embedding)*, producen salidas misma dimensión. El decodificador también incluye una pila de N = 6 capas idénticas. Además de las dos subcapas de cada capa del codificador, el decodificador inserta una tercera subcapa, que realiza la atención multi-cabezales sobre la salida de la pila del codificador. De manera similar al codificador, se emplea conexiones residuales alrededor de cada una de cada subcapa, seguidas de la normalización de las capas (Vaswani et al., 2017) [39].

Una función de atención se puede describir como la asignación de una consulta(Q) y un conjunto de pares clave-valor (K, V) a una salida, donde la consulta, las claves, los valores y la salida son todos vectores. La salida se calcula como una suma ponderada de los valores, donde el peso asignado a cada valor se calcula mediante una función de compatibilidad de la consulta con la clave correspondiente (Vaswani et al., 2017) [39].

En la Figura 3.3, se observa el esquema del mecanismo de atención, El objetivo principal de la atención es estimar la importancia relativa del término clave en comparación con el término de consulta relacionado con la misma persona o concepto . Para ello, el mecanismo de atención toma la consulta Q que representa una palabra vector, las claves K que son todas las demás palabras de la oración, y el valor V representa el vector de la palabra, V es igual a Q (para las dos capas de auto-atención). Al calcular el producto escalar normalizado entre la consulta y las claves, se obtiene un tensor que representa la importancia relativa de cada palabra para la consulta. Una palabra se representa mediante un vector en un espacio euclidiano, en este caso un vector de tamaño 512.

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

Cuando se calcula el producto escalar entre Q y KT, se calcula el producto entre la proyección ortogonal de Q en K. Es decir, que se trata de estimar cómo se alinean los vectores (las palabras entre la consulta y las claves) y regresa un peso para cada palabra de la oración.

Luego, se normaliza el resultado al dividir por la raíz del tamaño de K (tamaño de la secuencia). Es necesario para evitar problemas de fuga de gradiente que se

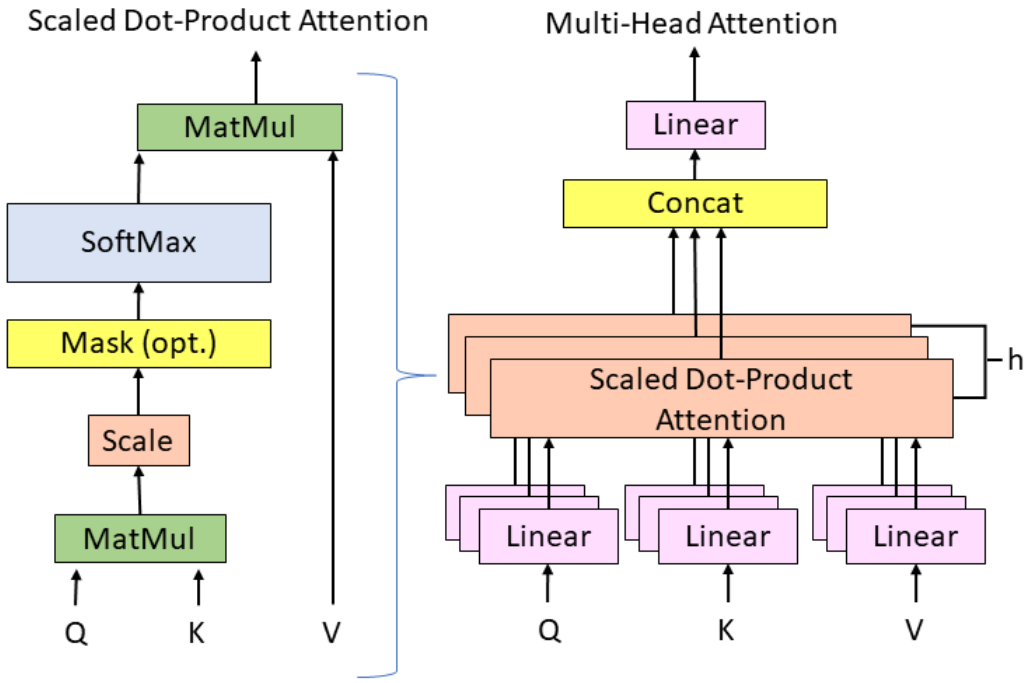

Figura 3.3: Atención por producto escalar(izquierda) y atención de multi-cabezal, consiste en varias capas de atención que se ejecutan en paralelo (derecha) (Vaswani et al., 2017). [39].

producirían en la función softmax si hay valores de gran tamaño. Se aplica la función softmax para intentar escalar el peso de la palabra en un rango entre 0 y 1. Al final, se multiplican estos pesos por el valor (el vector de la palabra) para reducir la importancia de palabras no relevantes y quedarse solo con las que mas importan.

El Transformer utiliza el mecanismo de Atención Multi-Head, es simplemente una proyección de Q, K y V en h espacios lineales. Siendo h la cantidad de cabezas que tiene el mecanismo (siendo $h = 8$ Vaswani et al., explica en su articulo [39]). Esto permite que cada cabeza se centre en aspectos diferentes, para después concatenar los resultados, permitiendo que la propia palabra no sea la dominante en el contexto.

Vaswani et al., compara en su articulo [39] la arquitectura de Transformer y otros modelos de última generación en 2017. La arquitectura Transformer supera a todos los modelos en la prueba BLEU, prueba evalúa el algoritmo en una tarea de

traducción. Comparó la diferencia entre la traducción proporcionada por el algoritmo y los humanos (Vaswani et al., 2017). [39]..

Los transformadores son un gran avance en NLP, superan a RNN al tener un menor costo de entrenamiento que permite entrenar modelos en corpus más grandes. Incluso hoy en día, los transformadores siguen siendo la base de modelos de última generación como BERT, Roberta, entre otros.

3.3.1. BERT

Modelo de representación de lenguaje BERT, *Bidirectional Encoder Representations from Transformers*, uno de los modelos más populares basados en Transformer, diseñado para entrenar previamente representaciones bidireccionales profundas a partir de texto sin etiquetar. Creado por los investigadores de Google, BERT es la primera representación de lenguaje no supervisada profundamente bidireccional, preentrenada usando solo un corpus de texto de Wikipedia sin formato (Devlin et al., 2018). [13].

Los autores Devlin et al., explican que las representaciones preentrenadas pueden ser contextuales o libres de contexto, las representaciones contextuales pueden ser unidireccionales o bidireccionales . Los modelos independientes del contexto, como Word2vec o GloVe, generan una representación incrustada (*embedding*) de una sola palabra para cada palabra del vocabulario. Por ejemplo, la palabra "*banco*"tendría la misma representación sin contexto en "*cuenta de banco*z "*banco del río*". En cambio, los modelos contextuales generan una representación de cada palabra que se basa en las otras palabras de la oración. Por ejemplo, en la oración "*Accedí a la cuenta del banco*", un modelo contextual unidireccional representaría "*banco*"basado en "*Accedí a*"pero no a "*cuenta*". Sin embargo, BERT representa "*banco*ütilizando tanto su contexto anterior como el siguiente: "*Accedí a la ... cuenta*", comenzando desde el fondo de una red neuronal profunda, haciéndola profundamente bidireccional (Devlin et al., 2018). [13]..

En la Figura 5.3 se muestra una visualización de la arquitectura de la red neuronal de BERT. Las flechas indican el flujo de información de una capa a la siguiente. Los cuadros verdes en la parte superior indican la representación contextualizada final de cada palabra de entrada.

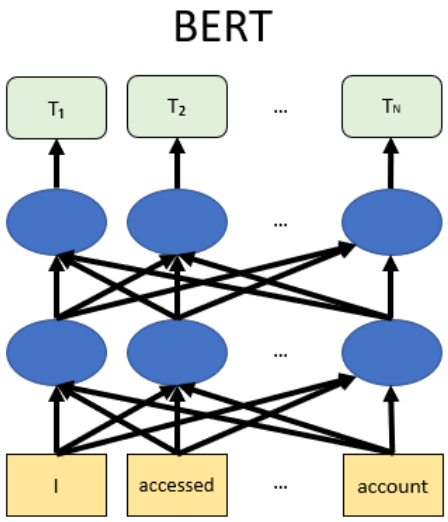

Figura 3.4: Arquitectura de la red neuronal de BERT profundamente bidireccional (Devlin et al., 2018). [13].

BERT hace uso de Transformer, un mecanismo de atención que aprende relaciones contextuales entre palabras (o subpalabras) en un texto. En su forma básica, Transformer incluye dos mecanismos separados: un codificador que lee la entrada de texto y un decodificador que produce una predicción para la tarea. Dado que el objetivo de BERT es generar un modelo de lenguaje, solo es necesario el mecanismo del codificador.

En modelos unidireccionales se entrenan eficientemente al predecir cada palabra condicionada a las palabras anteriores en la oración, Sin embargo, no es posible entrenar modelos bidireccionales simplemente condicionando cada palabra en sus palabras anteriores y siguientes, ya que esto permitiría que la palabra que se predice se vea reflejada indirectamente en un modelo de múltiples capas. Para resolver este problema se utiliza la técnica de enmascarar algunas de las palabras en la entrada y luego condicionar cada palabra bidireccionalmente para predecir las palabras enmascaradas, técnica que por la primera vez se usa con éxito para preentrenar una red neuronal profunda con BERT.

Antes de introducir secuencias de palabras en BERT, el 15 % de las palabras de cada secuencia se reemplazan con un token [MASK]. Luego, el modelo intenta

predecir el valor original de las palabras enmascaradas, en función del contexto proporcionado por las otras palabras no enmascaradas de la secuencia (Horev, 2018) [20]. En términos técnicos, la predicción de las palabras de salida requiere:

1. Agregar una capa de clasificación encima de la salida del codificador.
2. Multiplicando los vectores de salida por la matriz de incrustación, transformándolos en la dimensión del vocabulario.
3. Cálculo de la probabilidad de cada palabra en el vocabulario con softmax.

3.3.2. RoBERTa

RoBERTa, siglas en ingles de *A Robustly Optimized BERT Pretraining Approach*, enfoque de preentrenamiento BERT robustamente optimizado. Modelo propuesto por autores Liu et al, basado en el modelo BERT de Google lanzado en 2018, aunque se que puede igualar o superar la rendimiento de todos los métodos post-BERT (Liu et al., 2019) [29].

Roberta itera sobre el procedimiento de preentrenamiento de BERT, incluido el entrenamiento del modelo por más tiempo, con lotes más grandes sobre más datos; eliminar el siguiente objetivo de predicción de oraciones; entrenamiento en secuencias más largas; y cambiando dinámicamente el patrón de enmascaramiento aplicado a los datos de entrenamiento. También recopilamos un gran conjunto de datos nuevo (*CC-NEWS*) de tamaño comparable a otros conjuntos de datos de uso privado, para controlar mejor los efectos del tamaño del conjunto de entrenamiento (Liu et al., 2019) [29].

3.3.3. SciBERT

SciBERT es un modelo de lenguaje preentrenado basado en BERT para abordar la falta de datos científicos etiquetados a gran escala y de alta calidad. SciBERT aprovecha el pre entrenamiento no supervisado en un gran corpus de publicaciones científicas de múltiples dominios (*Semantic Scholar*) para mejorar el rendimiento en tareas científicas posteriores de PNL, creado por el *Allen Institute for Artificial Intelligence (AI2)* (Beltagy et al., 2019) [6]. SciBERT tiene su propio vocabulario,

SciVocab, creado para adaptarse mejor al corpus de entrenamiento, también incluye modelos entrenados en el vocabulario BERT original, *BaseVocab*, para comparar.

Los autores Beltagy et al. evaluaron SciBERT en un conjunto de tareas y conjuntos de datos de dominios científicos, lo cual superó significativamente a BERT y logra nuevos resultados de estado del arte en varias de estas tareas (Beltagy et al., 2019) [6].

3.4. Conditional Random Fields (CRF)

Conditional Random Fields (CRF), campos aleatorios condicionales, un método para construir modelos probabilísticos para segmentar y etiquetar datos de secuencia, adecuado para tareas de predicción donde la información contextual o el estado de los vecinos afectan la predicción actual. Los CRF encuentran sus aplicaciones en el reconocimiento de entidades nombradas, parte del etiquetado de voz, predicción de genes, reducción de ruido y problemas de detección de objetos, entre otros (Sutton et al., 2010) [36].

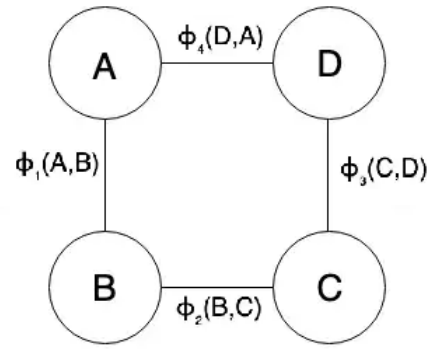

Figura 3.5: Grafo - MRF con cuatro variables aleatorias.

Los CRF una especie de campo aleatorio de Markov (MRF), una red de Markov es una clase de modelos gráficos con un gráfico no dirigido entre variables aleatorias. El autor Prasad, en su articulo, explica que a estructura de este gráfico decide la dependencia o independencia entre las variables aleatorias. Una red de Markov está

representada por un gráfico $G = (V, E)$ con los vértices o nodos que representan variables aleatorias y los bordes que representan colectivamente las dependencias entre esas variables (Prasad, 2019) [34].

$$Pr(A = a, B = b, C = c, D = d) = \frac{\phi_1(a,b)\phi_2(b,c)\phi_3(c,d)\phi_4(d,a)}{\sum_{a'}\sum_{b'}\sum_{c'}\sum_{d'}\phi_1(a',b')\phi_2(b',c')\phi_3(c',d')\phi_4(d',a')}$$

Figura 3.6: Probabilidad conjunta como producto normalizado de factores (Prasad, 2019) [34].

El gráfico se puede factorizar en J factores diferentes, cada uno de los cuales se rige por una función factorial Φ_j cuyo alcance es un subconjunto de variables aleatorias D_j. El $\Phi_j(d_j)$ debe ser estrictamente positivo para todos los valores posibles de d_j. Para que un subconjunto de variables aleatorias se represente como un factor, todas ellas deben estar conectadas entre sí en el gráfico. Además, la unión de alcances de todos los factores debe ser igual a todos los nodos presentes en el gráfico. La probabilidad conjunta no normalizada de las variables es el producto de todas las funciones factoriales, es decir, para el MRF que se muestra con $V = (A, B, C, D)$, la probabilidad conjunta se puede escribir como en siguiente ecuación 3.6.

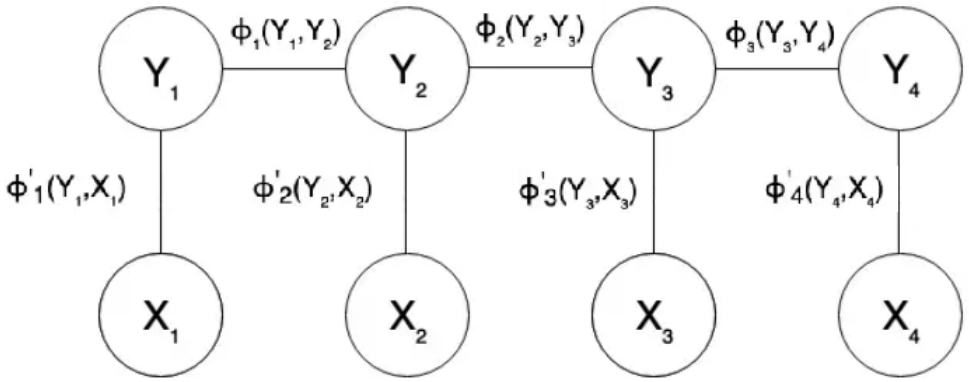

Figura 3.7: Estructura de campo aleatorio condicional (CRF).

El denominador es una suma del producto de factores sobre todos los valores po-

sibles que pueden tomar las variables aleatorias. Es una constante, también conocida como función de partición y comúnmente se denota por Z. Uno de esos grafo que satisface la propiedad de Markov es el grafo estructurado en cadena que se observa en la Figura 3.7.

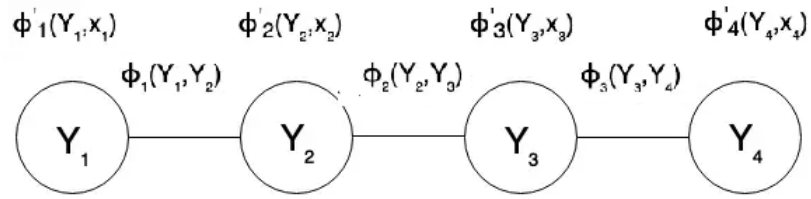

Figura 3.8: Modelo CRF, acondicionado en X.

Dado que CRF es un modelo condicional, modela la probabilidad condicional $P(Y/X)$, es decir, X siempre es dado o observado, por lo tanto, el grafo finalmente se reduce a una cadena simple 3.8.

Al acondicionar a X y se trata de encontrar el Y_i correspondiente para cada X_i, X e Y también se denominan variables de evidencia y etiqueta respectivamente. Se puede verificar que el modelo CRF, factor reducido, sigue la propiedad de Markov como se muestra para la variable Y_i a continuación y la probabilidad condicional de Y_2 dadas todas las demás variables finalmente depende solo de sus nodos vecinos, Figura 3.9.

$$
\begin{aligned}
P(Y_2/Y_1Y_3Y_4) &= \frac{P(Y_2Y_1Y_3Y_4)}{P(Y_1Y_3Y_4)} \\
&= \frac{\phi_1'(Y_1,x_1)\phi_1(Y_1,Y_2)\phi_2'(Y_2,x_2)\phi_2(Y_2,Y_3)\phi_3'(Y_3,x_3)\phi_3(Y_3,Y_4)\phi_4'(Y_4,x_4)}{\sum_{y_2'}\phi_1'(Y_1,x_1)\phi_1(Y_1,y_2')\phi_2'(y_2',x_2)\phi_2(y_2',Y_3)\phi_3'(Y_3,x_3)\phi_3(Y_3,Y_4)\phi_4'(Y_4,x_4)} \\
&= \frac{\phi_1(Y_1,Y_2)\phi_2'(Y_2,x_2)\phi_2(Y_2,Y_3)}{\sum_{y_2'}\phi_1(Y_1,y_2')\phi_2'(y_2',x_2)\phi_2(y_2',Y_3)}
\end{aligned}
$$

Figura 3.9: Variable Y_2 que satisface la propiedad de Markov (Prasad, 2019) [34].

Capítulo 4

Metodología

En esta investigación se considera la extracción de Cantidad (*Quantity (Q)*), Entidad Medida (*MeasuredEntity (ME)*), Propiedad Medida (*MeasuredProperty (MP)*), opcionalmente un Calificador (*Qualifier (Qr)*) y las relaciones *HasEntity, HasProperty, Qualifier*. Los valores también pueden extraer atributos adicionales de *Modifier* (modificadores de la cantidad) como *isMean, isApproximate, isCount,isRange, isList, isMedian*, entre otros.

Después de realizar la extracción de *Quantity*, otros atributos *(MeasuredEntity, Property, y Qualifier)* relacionados a esas cantidades requieren ser predecidos. La figura 4.1 da una visión general del método propuesto. El conjunto de aristas a cada nodo representa la entrada del modelo entrenado y la etiqueta de cada nodo representa la predicción realizada por el modelo.

Los datos de entrada son los proporcionados por OA-STM-Corpus (OA-STM-Corpus, 2017)[32] definido en el capitulo de conjunto de datos. Esto motivó el uso de BERT, SciBERT y RoBERTa para la obtención de las relaciones y extracción del etiquetado de cada elemento.

4.1. Conjunto de Datos

Los datos de entrada son los proporcionados por OA-STM-Corpus Elsevier Labs puso a disposición previamente. (OA-STM-Corpus, 2017) [32], es un repositorio de Corpus de artículos de acceso abierto de múltiples campos en ciencia, tecnología y

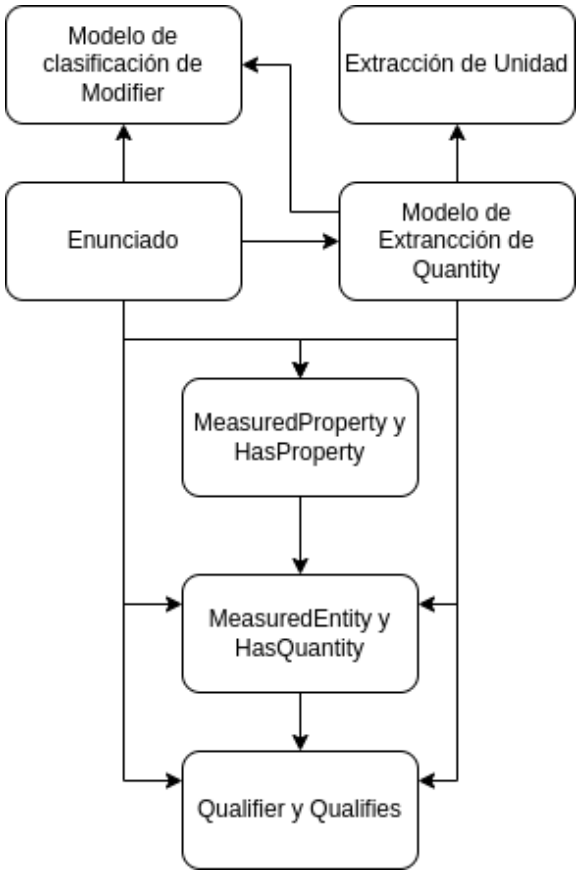

Figura 4.1: Modelo general de extracción de entidades y relaciones.

medicina.

Los artículos científicos del corpus de entrenamiento y de prueba pertenecen a los siguientes subdominios: Astronomía, Ingeniería, Medicina, Ciencia de los Materiales, Biología, Química, Agricultura, Ciencias de la Tierra e Informática. Estos artículos fueron etiquetados manualmente de mediante la Krippendorff.

Las anotaciones se extraen de 110 artículos con licencia CC-BY, que fueron la base de una tarea previa de SemEval para SemEval 2017 (Augenstein et al., 2017) [4]. Estos 110 artículos se distribuyen uniformemente en 10 áreas temáticas. De estos 110 artículos, el conjunto de datos MeasEval incluye 428 párrafos que contienen 1663 Cantidades. Estos se dividen en un conjunto de datos de entrenamiento de 1164 Cantidades (313 párrafos) y un conjunto de evaluación de 499 Cantidades (135 párrafos).

Todos los párrafos fueron anotados por al menos dos anotadores, luego revisados y reconciliados durante una reunión de adjudicación, a menudo incluyendo un tercer anotador. La publicación de datos de MeasEval incluyó datos de entrenamiento, así como anotaciones originales de múltiples anotadores para un subconjunto de 248 Cantidades de datos de entrenamiento. Esto fue para proporcionar información profunda sobre el acuerdo entre anotadores y también para permitir que los participantes hicieran su propio análisis sobre cómo funcionan sus algoritmos en relación con los humanos.

Los datos incluyen un archivo de texto para cada párrafo del texto científico, así como anotaciones que se encuentran en dos formatos. Las anotaciones se proporcionan en formato de archivo de valores separados por tabulaciones (.tsv), y también en formato de anotación BRAT. El formato BRAT es para fines de visualización y revisión, pero el formato oficial de los datos es el .tsv, que se utilizará para las presentaciones y la evaluación. Para los archivos .tsv y .txt se proporcionará un archivo por párrafo de texto anotado. Para los archivos BRAT, hay un conjunto adicional de 1 archivo .ann y 1 archivo .txt por Cantidad anotada.

Para este proyecto se utiliza el formato .tsv con los siguientes campos:

- **docId**: señala el ID del documento del ejemplo.
- **annotSet**: se refiere a la agrupación lógica de anotaciones, una por cantidad anotada, en el orden en que aparecen en el documento de texto.
- **annotType**: una de las siguientes opciones: Quantity, MeasuredEntity, MeasuredProperty o Qualifier.
- **startOffset**: desplazamiento de caracteres del inicio de la anotación en el texto.
- **endOffset**: desplazamiento de carácter que señala el carácter después del último carácter de la anotación.
- **annotId**: identificador de la fila en el archivo, único por annotSet.
- **text**: texto de la anotación.
- **other**: propiedades adicionales utilizadas en la tarea:

- Para Quantities: other contiene unit: la unidad en el texto; si: el equivalente SI de esta unidad, si procede, y mods: un conjunto de modificadores que describen con más detalle la Quantity.
- En el caso de MeasuredEntity, MeasuredProperty y Qualifier, other contiene el tipo de relación y el objetivo del tramo relacionado, de la forma {relationType: targetAnnotation}.

La anotación consta de Quantities, MeasuredEntities, MeasuredProperties y Qualifiers. Una cantidad puede ser un recuento o una medición y las mediciones se componen de una unidad y un valor.

Los valores también pueden tener atributos adicionales como *isMean*, *isApproximate* o *isRange*. Las cantidades pueden estar directamente relacionadas con una entidad medida o indirectamente a través de una propiedad medida. Los calificadores proporcionan información adicional necesaria para interpretar la medición. Por ejemplo, la presión a la que se ha observado el punto de ebullición o la profundidad y el lugar donde se ha tomado una muestra del océano.

Dado que los textos pueden contener distintas partes de esta información, todas las relaciones son opcionales. Una MeasuredEntity puede estar relacionada con una MeasuredProperty o una Cantidad, una MeasuredProperty puede estar relacionada con una Cantidad, y un Calificador puede tener relación con cualquier *span* (tramo).

Por ejemplo en el articulo de Kender [23], que hace parte de la base de datos, contiene el siguiente texto en formato en texto plano, al etiquetar queda como demostra la Figura 4.2, y cuenta con un archivo asociado tsv el cual cuenta con un etiquetado de esta forma:

"*Carbon isotopic results of total organic matter (δ 13CTOC) and amorphous organic matter (δ13CAOM) against core 22/10a-4 lithology and Apectodinium spp. (%). Blue=bulk rock δ13CTOC; black=δ13CAOM; solid red symbols=bulk rock δ13C TOC from samples with <30 % wood/plant tissue (determined from palynological residue of the sample); open red symbols=bulk rock δ 13CTOC from samples with >30 % wood/plant tissue. The first appearance of Apectodinium augustum identifies the PETM in the North Sea (Bujak and Brinkhuis, 1998), and the first negative shift in δ13C identifies the approximate position of the CIE onset and the Paleocene–Eocene boundary. Values shaded at 2614.7 and 2619.6 m are considered pos-*

sible outliers based on statistical analysis of the palynological residues (see Section 4.1). Lithologic column shows position of sand intervals (yellow), claystone intervals (brown; predominantly laminated claystone, dark brown), and ash layers (pink). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.). "(Kender,2012)[23].

docId	annotSet	annotType	startOffset	endOffset	annotId	text	other
S0012821X12004384-952	1	Quantity	249	253	T1-1	<30%	{"mods": ["IsRange"], "unit": "%"}
S0012821X12004384-952	1	MeasuredProperty	254	271	T2-1	wood/plant tissue	{"HasQuantity": "T1-1"}
S0012821X12004384-952	1	MeasuredEntity	236	243	T3-1	samples	{"HasProperty": "T2-1"}
S0012821X12004384-952	2	Quantity	380	384	T1-2	>30%	{"mods": ["IsRange"], "unit": "%"}
S0012821X12004384-952	2	MeasuredProperty	385	402	T3-2	wood/plant tissue	{"HasQuantity": "T1-2"}
S0012821X12004384-952	2	MeasuredEntity	367	374	T2-2	samples	{"HasProperty": "T3-2"}
S0012821X12004384-952	3	Quantity	658	677	T1-3	2614.7 and 2619.6 m	{"mods": ["IsList"], "unit": "m"}
S0012821X12004384-952	3	MeasuredEntity	641	654	T2-3	Values shaded	{"HasQuantity": "T1-3"}

Figura 4.2: Archivo de un documento en formato TSV con el etiquetado correspondiente.

4.2. Modelo de desarrollo

La arquitectura general del modelo se muestra en la Figura 4.3 en la cual se observa como es ingresada la información en formato CLS para posteriormente ser tokenizada en todos los casos por medio de SciBERT, posteriormente se ingresa a una red neuronal lineal completamente conectada con función Tahn (función de activación, con valores entre -1 y 1), se concatena el resultado de esta para obtener mayor capacidad de entrenamiento, esto significa que diversos enunciados son utilizados en la etapa de iteraciones de cada época de entrenamiento, por decirlo de otra forma cada instancia cuenta con mayor cantidad de elementos al momento de entrenamiento, debido a que se basa en el modelo en la herramienta BERT, con un máximo de 512 tokens ingresados al modelo para su entrenamiento. Posteriormente se implementa la segmentación de un párrafo en enunciados y estos enunciados son ingresados al modelo implementado (BERT, SciBERT, RoBERTa)

4.2.1. Extracción de Quantity

Los enunciados de entrada son tokenizados utilizando cada versión del modelo (BERT, SciBERT, RoBERTa) a partir de la implementación de HugginFace ***Wolf

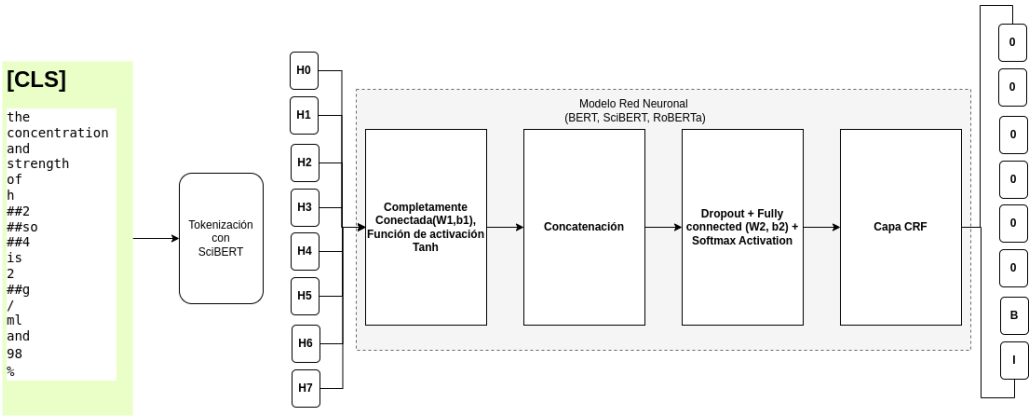

Figura 4.3: Modelo de la red neuronal para la extracción de relaciones.

et al. 2020). Los span de Quantity son transformados en formato BIO/IO y utilizados como un etiquetado de verdad durante el entrenamiento del modelo.

El enunciado tokenizado es ingresado al modelo SciBERT. La función de activación dedicada para la capa final fue la tanh

$$H_i = W_1[tanh(H_i)] + b_1 \text{ y } i = 0, 1, ..., len$$

Donde cada H_i es la unidad oculta correspondiente al token i y len es el tamaño máximo del enunciado tokenizado, de manera similar se procesa los tokens[CLS]:

$$H_{cls} = W_0[tanh(H_0)] + b_0$$

Finalmente se obtiene la representación final de la oración al concatenar H_{cls} y H_i y esta es utilizada para la predicción por medio de la función softmax.

$$H_i^{''} = W_2[concat(H_i^{'}, H_{cls}^{'})] + b_2 \quad i = 0, 1, ..., len$$
$$H^{''} = \left[H_0^{''}, H_1^{''}, ..., H_{len}^{''}\right]^T$$
$$p = softmax(H^{''},\ dim = -1)$$

Figura 4.4: Formula que concatena los elementos de la oración con sus etiquetados.

Las matrices W_0 y W_1 tienen la misma dimensión por lo que d es el tamaño del estado oculto de BERT y t representa el numero de etiquetas (3) en este caso porque se esta utilizando la codificación BIO.

CRF permite extraer dependencias estructurales entre las etiquetas BIO. El vector de probabilidad de etiqueta para todos los tokens, la p se pasa a través de la capa CRF para generar la secuencia de salida más probable. Se entrena el modelo calculando la perdida CRF y el optimizador Adam.

4.2.2. Extracción de Unidad

Las oraciones de cantidad son tokenizadas utilizando el tokenizador basado en caracteres de *Spacy*. El vector de etiquetas de verdad es entrenado y toma un formato de un vector binario indicando los índices en el span de unidades para las oraciones de cantidad.

Se entreno el modelo basado en caracteres Bi-LSTM con emmbeddings de palabras utilizando *loss* BCE(Entropia Binaria Cruzada por sus siglas en ingles) y el optimizador Adam, como se observa en la figura 4.5 la arquitectura del modelo.

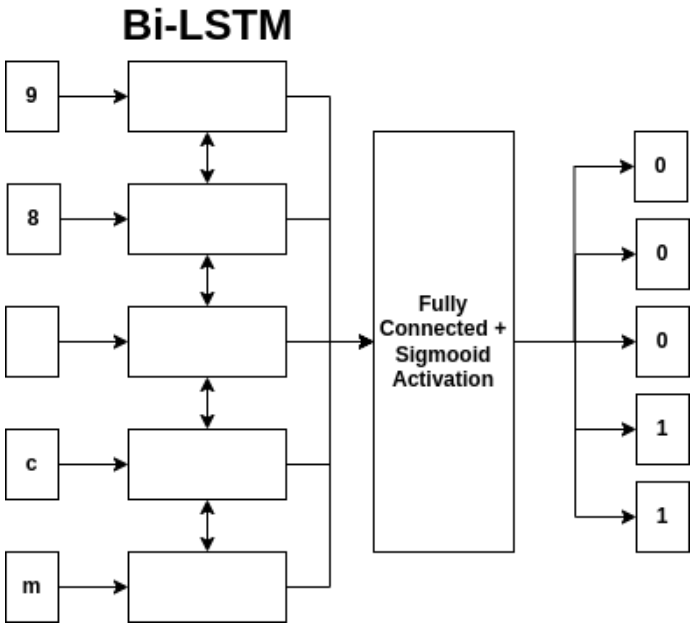

Figura 4.5: Modelo generado para la extracción de unidades utilizando Bi-LSM

4.2.3. Extracción del Modificador

Se implemento un modelo que extrae las relaciones del modificador de cantidades con las siguientes etiquetas:

- HasTolerance
- IsApproximate
- IsCount
- IsList
- IsMean
- IsMean-HasSD
- IsMeanHasTolerance
- IsMeanIsRange
- IsMedian
- IsRange
- IsRangeHasTolerance
- None

Donde cada una de ellas representa el tipo de relación que se tiene con Quantity.

Para que los modelos basados en BERT fueran capaces de capturar la ubicación de Quantity se inserto un símbolo de $ al principio y fin de cada span de Quantity. Si hay varias cantidades en una oración, se generan varias copias de la misma oración con el símbolo $ en diferentes posiciones. Supone que H_i hasta H_j es el vector final de estados ocultos para el span de Quantity, entonces la operación promedio es aplicada al vector.La salida entonces es comunicada a través de una capa completamente conectada seguida por una activación Sigmoid. Como se observa en la figura 4.6.

El modelo fue entrenado utilizando BCE (Entropia Binaria Cruzada por sus siglas en ingles) y el optimizador Adam. El valor umbral para la predicción se determinó mediante validación cruzada.

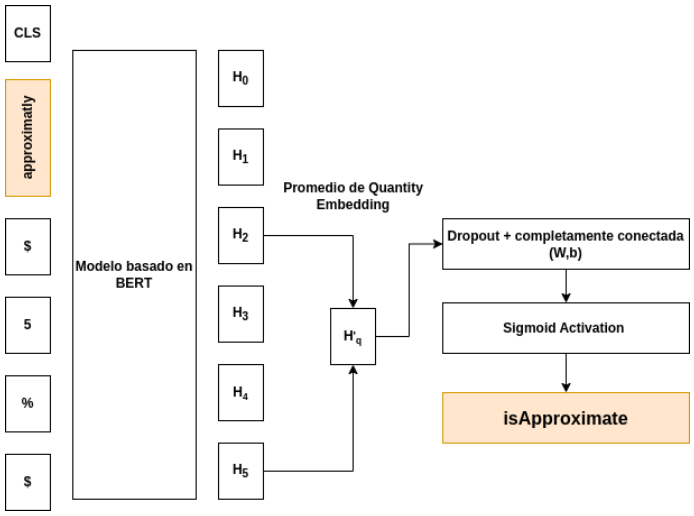

Figura 4.6: Modelo generado para la extracción de modificadores utilizando la activación Sigmoide.

4.2.4. Extracción de MeasuredEntity y HasQuantity

Así como se realizo en la clasificación anterior para capturar la ubicación, se inserta el símbolo de $ al principio y al final del span de Quantity. Las oraciones modificadas son tokenizadas utilizando el tokenizador de SciBERT.

El span de MeasuredEntity que se encuentra relacionado con Quantity encapsulado entre los $ es transformado en el formato BIO/IOB y es utilizado como la etiqueta de verdad para entrenar el modelo.

Los datos con el formato definido son utilizados para entrenar el modelo de manera similar al la extracción de Quantity. El modelo extrae MeasuredEntity asociado con el Quantity por lo que predice las relaciones de MeasuredEntity y al mismo tiempo HasQuantity del MeasuredEntity predecido.

4.2.5. Extracción de MeasuredProperty y HasProperty

Para realizar la clasificación y extracción de las relaciones de MeasuredProperty y HasProperty se utiliza en enfoque similar al utilizado en la fase anterior, se encapsula el Quantity span entre los símbolos de $ y la MeasuredEntity span dentro del símbolo de #. Las oraciones modificadas son ingresadas al tokenizador de SciBERT. El span de MeasuredProperty relacionado con el par MeasuredEntity y Quantity es

transformado en el formato BIO/IOB y utilizado como etiqueta de verdad para el entrenamiento del modelo.

Los datos con el formato son utilizados para entrenar el modelo de manera similar al de la extracción de Quantity (Modelos Basados en BERT + CRF), el modelo entrenado es utilizado para extraer MeasuredProperty ligado con el par MeasuredEntity y Quantity. Si el modelo predice un span de MeasuredProperty entonces la relación HasQuantity es actualizada a MeasuredProperty y la relación HasProperty es agregada a MeasuredEntity.

4.2.6. Extracción de Qualifier y Qualifies

Para extraer los spans de Qualifier y Qualifies dos bloques son utilizados. Mientras se entran el primer bloque se inserta $ al principio y al final del span de Quantity esto debido a que se asume que Qualifier califica a Quantity. Durante el segundo entrenamiento se encapsula el span de MeasuredProperty en $ debido a que se asume que Qualifier califica a Measuredproperty.

4.2.7. Post procesamiento

Una vez que las predicciones de cada modelo se encuentran disponibles requerimos de transformar el formato BIO/IOB en un formato de entidad de span, inicialmente se mapea cada span del token en una oración tokenizada y se utiliza para determinar el span de la entidad predecida. Mientras que si se encuentra multiples entidades asociadas a los span de MeasuredEntity, MeasuredProperty o Qualifier se selecciona la que se encuentra mas cerca del Quantity span. Posteriormente se convierte el span de la oración de cada entidad extraída del párrafo de span.

4.3. Experimentos

Se realizan los experimentos con el conjunto de datos segmentado en 80 % para entrenamiento y 20 % para validación.

Los parámetros de la red neuronal fueron definidos de la siguiente manera:

- Estados ocultos: 768

- Dropout: 0.1
- Batch Size: 24
- Learning Rate: 0.00001
- Threshold: 0.5

El valor de los estados ocultos se determina en 768 ya que es la compatibilidad máxima para el ingreso del vector dimensional de la red, aunque para RoBERTa la dimensión máxima soportada es de 1024 se eligió permitir que los modelos fueran comparados con la mayor igualdad posible para garantizar que el análisis fuera fidedigno.

El valor de Dropout permite obtener resultados de entrenamiento que ayudan a que la red no sea sobreentrenada y generalice de una forma mejor. Con el 0.1 de Dropout garantizamos que en cada iteración de la época sean desechados el 10 % de los instancias logrando así la diversidad dentro de las muestras analizadas en cada iteración.

Para el Batch Size se trato de utilizar diversas dimensiones pero se encontró que con bloques muy grandes de información el sistema entregaba resultados erráticos esto debido a la limitación del ingreso de la red basada en alguna versión de BERT. Cuando el lote es mayor por ejemplo de 64 las oraciones ingresadas sobrepasan el tamaño máximo de entrada del vector denso entonces muchos elementos son ignorados y no se analizan al momento de realizar el proceso de calculo en la capa totalmente conectada. Si el Batch es mas pequeño digamos 12 el sistema tiene un comportamiento similar pero tarda significativamente mas en converger debido a que tiene que analizar muchas mas veces pocos elementos.

Cuando un modelo nos da una puntuación en lugar de la predicción en sí, normalmente se necesita convertir esta puntuación en una predicción aplicando un Threshold. Dado que el significado de la puntuación es dar la probabilidad percibida de tener 1 según el modelo, es obvio utilizar 0.5 como umbral. De hecho, si la probabilidad de tener 1 es mayor que la de tener 0, es natural convertir la predicción en 1. 0.5 es el umbral natural que asegura que la probabilidad dada de tener 1 es mayor que la probabilidad de tener 0. Por eso es el umbral por defecto utilizado en la librería scikit-learn de Python cuando se llama al método predict de una instancia

de estimador. Reduciremos el ruido de las predicciones aplicándoles un umbral 0.5. Se toma sólo las predicciones de etiquetas superiores (o iguales) al umbral.

Se realizaron experimentos con iteraciones de hasta 50 épocas encontrando que aproximadamente a partir de la época 25 el sistema se estabiliza.

Los entrenamientos se realizaron en un equipo con las siguientes características:

- Procesador Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
- 128 GB de Memoria RAM DDR4 ECC a 2400 Mhz.
- 2 TB de Almacenamiento en Disco SSD
- 8 Tarjetas Gráficas Nvidia GTX 1080 Ti con 11 GB de RAM DDR5 y 3584 núcleos CUDA cada una de ellas.

Capítulo 5

Resultados

Las métricas utilizadas fueron las mismas de SemEval 2021 task8 *F1-measure*, *F1-overlap* y *Exact Match. Exact Match* es un valor binario de 0 o 1, mientras que *F1-measure* es una relación de solapamiento a nivel de token de los espacios de presentación a los verdaderos, donde la tokenización se realiza utilizando delimitadores simples de espacio en blanco. El *F1-overlaping* esta basada en la *F1-measure* que penaliza más estrictamente los envíos negativos. La evaluación final se basa en una puntuación de *F1-overlap* global promediada en todas las subtareas.

Aquí el valor mas importante es el *F1-overlap* ya que da un valor funcional de comportamiento general del sistema en todas las tareas. La cual esta definida en la Fig 5.1 en donde se establece que el valor de F1 estará asociado a la exactitud y al recuerdo, donde *Precision* es el acumulado de exactitud de cada una de las tareas y *Recall* es el acumulado de recuerdo de cada uno de las tareas.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Figura 5.1: Formula utilizada para el calculo de F1 en general para todas las tareas.

5.1. Resultados de Extracción de Cantidad

Los resultado de la extracción de cantidad se muestran a continuación, como podemos observar la extracción del elemento de cantidad como aparecen en las Figuras 5.1 5.2 y 5.3 en todos los casos fue bastante exitosa, este es el subproceso mas sencillo

debido a que es básicamente la extracción de un elemento numérico que se encuentra muy asociado a las tareas tokenizadoras, es así como casi todos los participantes en la tarea de SEMEVAL Task8 2021 tuvieron resultados sobresalientes en este tema en especifico.

Modelo SciBERT

Reconocimiento de Entidad Accuracy:-0.9038461538461539	
———Resultados NER———	
Accuracy:	**-0.99875**
Modified Accuracy:	**-0.9038461538461539**
Precision:	**-0.9170731707317074**
Recall:	**-0.9842931937172775**
F1 score:	**-0.9494949494949495**

Cuadro 5.1: Resultados del modelo implementado con SciBERT

Modelo RoBERTa

Reconocimiento de Entidad Accuracy:-0.853812346132149	
———Resultados NER———	
Accuracy:	**-0.95335**
Modified Accuracy:	**-0.853812346132149**
Precision:	**-0.874313421074**
Recall:	**-0.944234112321334**
F1 score:	**-0.90323332331284715**

Cuadro 5.2: Resultados del modelo implementado con RoBERTa

5.2. Resultados individuales en cada elemento y F1

En esta sección podemos observar en las figuras 5.4 y 5.5 como es que el modelo entrenado con SciBERT tuvo en casi todos los casos un mejor comportamiento al momento de extraer las relaciones mas complejas del proyecto, el motivo por el cual se detecta este comportamiento es que SciBERT es entrenado con artículos científicos de diversas areas las cuales en su mayoría se yuxtaponen con los tópicos

Modelo BERT

Reconocimiento de Entidad Accuracy:-0.8823529411764706	
———Resultados NER———	
Accuracy:	**-0.9985**
Modified Accuracy:	**-0.8823529411764706**
Precision:	**-0.9326424870466321**
Recall:	**-0.9424083769633508**
F1 score:	**-0.9374999999999999**

Cuadro 5.3: Resultados del modelo implementado con BERT

de los artículos con los que fue entrenado este proyecto y que se encuentran en el repositorio de la tarea de SEMEVAL task8 2021.

La sub tarea que en general se asocia con un complejidad muy grande es la de *Qualifies* 5.5 esto debido a que en muchos textos se carece de ella y en los que llega a existir es una relación bastante compleja de identificar por cualquier medio y en nuestro caso por la red neuronal. Algo similar sucede con la subtarea que involucra la obtención de *Qualifier* donde nuestro sistema en todos los modelos implementados no obtuvo buenos resultados.

En general el valor de *F1-Overlap* se encontró próximo a los resultados obtenidos por los participantes del SEMEVAL task8 2021, este valor es el obtenido del calculo ponderado de todos los demás valores y se obtiene por medio de la herramienta ofrecida en el repositorio de la tarea mencionada, por lo que identificamos a los modelos basados en BERT como una herramienta útil, practica y eficiente en el desarrollo de aplicaciones que tratan con textos y a SciBERT como la herramienta que genero mejores resultados en este caso por ser orientada a textos científicos.

Modelo	Conjunto de Datos	Quantity	Unit	Modifier	MeasuredEntity	MeasuredProperty
BERT	Entrenamiento	0.861	0.774	0.614	0.302	0.216
	Validación	0.850	0.770	0.602	0.303	0.215
SciBERT	Entrenamiento	**0.878**	**0.807**	**0.696**	**0.406**	0.245
	Validación	0.859	0.800	0.670	0.398	**0.255**
RoBERTa	Entrenamiento	0.874	0.675	0.379	0.322	0.163
	Validación	0.849	0.650	0.350	0.311	0.145

Cuadro 5.4: Resultados 1 de 2

Modelo	Conjunto de Datos	Qualifier	HasQuantity	HasProperty	Qualifies	F1-Overlap
BERT	Entrenamiento	**0.083**	0.193	0.114	0.064	0.410
	Validación	0.080	0.188	0.117	0.076	0.405
SciBERT	Entrenamiento	0.077	**0.311**	**0.183**	**0.083**	**0.432**
	Validación	0.078	0.300	0.150	0.070	0.428
RoBERTa	Entrenamiento	0.080	0.270	0.137	0.070	0.330
	Validación	0.076	0.234	0.138	.067	0.327

Cuadro 5.5: Resultados 2 de 2

Conclusiones

En este trabajo se exploraron diversos modelos para la identificación y extracción de cantidades, medidas, unidades, atributos de las cantidades y propiedades, así como relaciones de conteo y de medidas en textos de discursos científicos, basándonos en los requerimientos, especificaciones y métricas del proyecto SEMEVAL Task8 2021 en el cual se trata de obtener las relaciones que existen entre los valores numéricos sus métricas y relaciones con otros elementos en el texto de documentos científicos.

Los resultados que se obtuvieron muestran que el funcionamiento de los modelos implementados se encuentra dentro del promedio de los concursantes del proyecto,

Algunas de las mejoras que se pueden aplicar al proyecto seria la implementación de un conjunto de datos mucho mas grande, por lo general las redes neuronales que se encuentran asociadas a los temas de procesamiento del lenguaje natural tiene un comportamiento mucho mejor y logran generalizar de forma correcta cuando los datos de entrada son masivos, esto seria, crear un conjunto de datos con al menos 10,000 documentos por área para así tener un sistema que entrene con datos mucho mas robustos. Un problema que se detecto en los documentos al analizarlos manualmente es que algunas incidencias de los valores aparecían en formatos muy específicos de acuerdo al área por lo que entre mas datos se pudieran obtener para tener una muestra poblacional mucho mas robusta seria de gran ayuda para la red neuronal.

Bibliografía

[1] Esther Omolara Abiodun, Abdulatif Alabdulatif, Oludare Isaac Abiodun, Moatsum Alawida, Abdullah Alabdulatif, y Rami S. Alkhawaldeh. Natural language processing. *Neural Comput Applic*, 33:15091–15118, 2021. ISSN 2347-2693. doi:https://doi.org/10.1007/s00521-021-06406-8. URL https://link.springer.com/article/10.1007/s00521-021-06406-8.

[2] Vraj Shah Aditya Jain, Gandhar Kulkarni. Natural language processing. *International Journal of Computer Sciences and Engineering*, 6:161–167, 2018. ISSN 2347-2693. doi:https://doi.org/10.26438/ijcse/v6i1.161167. URL https://www.ijcseonline.org/full_paper_view.php?paper_id=1652.

[3] Utkarsh Ankit. *Built In - Artificial Intelligence Articles.* 2022. Acceso: 15.11.2022. URL https://universoabierto.org/2018/02/22/que-es-la-mineria-de-textos-como-funciona-y-por-que-es-util/.

[4] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, y Andrew McCallum. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, págs. 546–555. Association for Computational Linguistics, Vancouver, Canada, 2017. doi: 10.18653/v1/S17-2091. URL https://aclanthology.org/S17-2091.

[5] Andrei-Marius Avram, George-Eduard Zaharia, Dumitru-Clementin Cercel, y Mihai Dascalu. UPB at SemEval-2021 task 8: Extracting semantic information on measurements as multi-turn question answering. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 534–

540. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.65. URL https://aclanthology.org/2021.semeval-1.65.

[6] Iz Beltagy, Kyle Lo, y Arman Cohan. Scibert: A pretrained language model for scientific text. 2019. doi:10.48550/ARXIV.1903.10676. URL https://arxiv.org/abs/1903.10676.

[7] Néstor Camilo Beltrán Beltrán y Edda Camila Rodríguez Mojica. Procesamiento del lenguaje natural (pln) - gpt-3.: Aplicación en la ingeniería de software. *Tecnología Investigación y Academia*, 8(1):18–37, 2021. ISSN 2344-8288. URL https://revistas.udistrital.edu.co/index.php/tia/article/view/17323.

[8] Jiarun Cao, Yuejia Xiang, Yunyan Zhang, Zhiyuan Qi, Xi Chen, y Yefeng Zheng. CONNER: A cascade count and measurement extraction tool for scientific discourse. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 1239–1244. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.176. URL https://aclanthology.org/2021.semeval-1.176.

[9] Amed Abel Castillo Zayas, Mariela; Leiva Mederos. La minería de texto: perspectiva metodológica para la realización de resúmenes documentales / text mining: a methodological perspective for document summaries. *ACIMED*, 15(05):161–167, 2015. doi:10.26438/ijcse/v6i1.161167.

[10] Jiakang Chang, Christian O' Reilly, Nancy Pontika, Gareth Owen, Kenneth Haug, y Martine Oudenhoven. ¿qué es la minería de textos, cómo funciona y por qué es útil? *Universo Abierto. Blog de la biblioteca de Traducción y Documentación de la Universidad de Salamanca.* 2018. Acceso: 03.10.2022. URL https://universoabierto.org/2018/02/22/que-es-la-mineria-de-textos-como-funciona-y-por-que-es-util/.

[11] Adis Davletov, Denis Gordeev, Nikolay Arefyev, y Emil Davletov. LIORI at SemEval-2021 task 8: Ask transformer for measurements. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*,

págs. 1249–1254. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.178. URL https://aclanthology.org/2021.semeval-1.178.

[12] IIC Instituto de Ingeniería del Conocimiento. Procesamiento del lenguaje natural. 2022. Acceso: 01.09.2022. URL https://www.iic.uam.es/inteligencia-artificial/procesamiento-del-lenguaje-natural/.

[13] Jacob Devlin y Ming-Wei Chang. Open sourcing bert: State-of-the-art pre-training for natural language processing. *Google Research. Natural Language Understanding.* 2018. Acceso: 09.11.2022. URL https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, y Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. En *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, págs. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, 2019. doi:10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[15] Ricardo Eíto Brun y José Antonio Senso. Minería textual. *EPI El Profesional de la información - Revista internacional científica y profesional.*, 13(1), 2004. ISSN 1386-6710. URL http://eprints.rclis.org/11491/1/Artmineriapdf.pdf.

[16] Akash Gangwar, Sabhay Jain, Shubham Sourav, y Ashutosh Modi. Counts@IITK at SemEval-2021 task 8: SciBERT based entity and semantic relation extraction for scientific data. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 1232–1238. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.175. URL https://aclanthology.org/2021.semeval-1.175.

[17] Isidoro Gil Leiva y José Vicente Rodríguez Muñoz. El procesamiento del lenguajes natural aplicado al análisis del contenido de los documentos. *Revista general*

de información y documentación, 6(2), 1996. ISSN 1132-1873, ISSN-e 1988-2858. URL https://dialnet.unirioja.es/servlet/articulo?codigo=169971.

[18] Corey Harper, Jessica Cox, Curt Kohler, Antony Scerri, Ron Daniel Jr., y Paul Groth. SemEval-2021 task 8: MeasEval – extracting counts and measurements and their related contexts. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 306–316. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.38. URL https://aclanthology.org/2021.semeval-1.38.

[19] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, y Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4), 2021. ISSN 0360-0300. doi:10.1145/3447772. URL https://doi.org/10.1145/3447772.

[20] Rani Horev. Bert explained: State of the art language model for nlp. *TDS. Towards Data Science.* 2018. Acceso: 16.11.2022. URL https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

[21] Kyle Hundman y Chris A. Mattmann. Measurement context extraction from text: Discovering opportunities and gaps in earth science. *CoRR*, abs/1710.04312, 2017. URL http://arxiv.org/abs/1710.04312.

[22] Neel Karia, Ayush Kaushal, y Faraaz Mallick. KGP at SemEval-2021 task 8: Leveraging multi-staged language models for extracting measurements, their attributes and relations. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 387–396. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.46. URL https://aclanthology.org/2021.semeval-1.46.

[23] S. Kender, M.H. Stephenson, James Riding, Melanie Leng, Robert Knox, Victoria Peck, Christopher Kendrick, Michael Ellis, Christopher Vane, y Rachel

Jamieson. Marine and terrestrial environmental changes in nw europe preceding carbon release at the paleocene–eocene transition. *Earth and Planetary Science Letters*, 353:108–120, 2012. doi:10.1016/j.epsl.2012.08.011.

[24] Natalia Konstantinova. Review of relation extraction methods: What is new out there? En *International Conference on Analysis of Images, Social Networks and Texts*, págs. 15–28. Springer, 2014.

[25] Hobson Lane y Maria Dyshel. *Natural Language Processing in Action*. Manning Early Access Program (MEAP), 2 ed$^{\text{ón}}$., 2021. ISBN 9781617299445. URL https://www.manning.com/books/natural-language-processing-in-action-second-edition.

[26] Nihatha Lathiff, Pavel PK Khloponin, y Sabine Bergler. CLaC-np at SemEval-2021 task 8: Dependency DGCNN. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 404–409. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.48. URL https://aclanthology.org/2021.semeval-1.48.

[27] George Lawton. What do nlp benchmarks like glue and squad mean for developers? *Search Enterprisea AI by TechTarget*, 2020. URL https://www.techtarget.com/searchenterpriseai/feature/What-do-NLP-benchmarks-like-GLUE-and-SQuAD-mean-for-developers.

[28] Patrick Liu, Niveditha Iyer, Erik Rozi, y Ethan A. Chi. Stanford MLab at SemEval-2021 task 8: 48 hours is all you need. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 1245–1248. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.177. URL https://aclanthology.org/2021.semeval-1.177.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, y Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

[30] Minh-Thang Luong, Hieu Pham, y Christopher D. Manning. Effective approaches to attention-based neural machine translation. 2015. doi:10.48550/ARXIV.1508.04025. URL https://arxiv.org/abs/1508.04025.

[31] Melikka Khosh Niat. Introduction to ie with gate based on material from hamish cunningham, kalina bont. *University of Sheffield*, 2010. URL https://www.is.inf.uni-due.de/courses/ie_ws10/folien/GATE-1_1.pdf.

[32] Elsevier Labs OA-STM-Corpus. Open access corpus of scientific, technical, and medical content, data based on cc-by sciencedirect articles. 2017. URL https://github.com/elsevierlabs/OA-STM-Corpus.

[33] Amir Pouran Ben Veyseh, Franck Dernoncourt, y Thien Huu Nguyen. DPR at SemEval-2021 task 8: Dynamic path reasoning for measurement relation extraction. En *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, págs. 397–403. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.47. URL https://aclanthology.org/2021.semeval-1.47.

[34] Aditya Prasad. Conditional random fields explained. 2019. URL https://towardsdatascience.com/conditional-random-fields-explained-e5b8256da776.

[35] Eduardo Sosa. Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones (parte i y ii). *EPI El Profesional de la información - Revista internacional científica y profesional.*, 6(1-2), 1996. ISSN 1386-6710. URL http://profesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html.

[36] Charles Sutton y Andrew McCallum. An introduction to conditional random fields. 2010. doi:10.48550/ARXIV.1011.4088. URL https://arxiv.org/abs/1011.4088.

[37] Benjamin Therien, Parsa Bagherzadeh, y Sabine Bergler. CLaC-BP at SemEval-2021 task 8: SciBERT plus rules for MeasEval. En *Proceedings of the 15th*

International Workshop on Semantic Evaluation (SemEval-2021), págs. 410–415. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.49. URL https://aclanthology.org/2021.semeval-1.49.

[38] Jakob Uszkoreit. Transformer: A novel neural network architecture for language understanding. *Google Research. Natural Language Understanding.* 2017. Acceso: 09.11.2022. URL https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, y Illia Polosukhin. Attention is all you need. En I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, y R. Garnett, eds., *Advances in Neural Information Processing Systems*, tomo 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.