

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**ANÁLISIS DE SENTIMIENTO EN INFORMACIÓN DE MEDIOS
PERIODÍSTICOS Y REDES SOCIALES MEDIANTE REDES
NEURONALES RECURRENTE**

**Tesis para obtener el título profesional de Ingeniero Informático que
presenta el bachiller:**

AUTOR:

Gabriel Hélar Zárte Calderón

ASESOR:

Dr. César Armando Beltrán Castañón

Lima, noviembre, 2021

Resumen

El análisis de sentimiento es un área de investigación importante en el procesamiento de lenguaje natural, la cual está en constante crecimiento especialmente por la generación de grandes volúmenes de información textual, y el avance tecnológico en lo que se refiere al almacenamiento y los algoritmos inteligentes para el análisis de esta. Esta tarea cada vez va tomando más fuerza su uso en diferentes aplicaciones computacionales dado el crecimiento exponencial del uso de medios digitales y redes sociales, las cuales, gracias a la información debidamente procesada, pueden ser muy valiosas para los negocios.

Actualmente existen procedimientos ambiguos para la realización de dicha tarea y sobre todo para textos en español y de manera específica para notas periodísticas y publicaciones realizadas en redes sociales, todo ello por el hecho de la escasa cantidad de herramientas existentes para la presente tarea, por ende el proceso de clasificación de las polaridades de los sentimientos expresadas en los textos se realiza de manera manual por expertos en el tema, generándose así resultados ambiguos y sesgados según la experiencia del encargado, lo cual generaba resultados que no eran del todo fiables retándole valor a dicha tarea, además del hecho de que realizarlo de manera totalmente manual resultaba muy pesado y se realizaba en un periodo largo de tiempo.

Para la realización de dicha tarea existen múltiples técnicas de aprendizaje de máquina y de aprendizaje profundo que son adecuadas para este, pero en el último año uno de los modelos que va siendo reconocido cada vez más para ser aplicado a resolver problemas de procesamiento de lenguaje natural son los modelos basados en *transformers* dadas sus buenas capacidades y los resultados que se obtienen con estos.

Ante dicha problemática surge la necesidad de investigar más acerca de cómo se vienen implementando soluciones para la realización de análisis de sentimiento para hacer una comparativa sobre los modelos usados y además dadas las buenas capacidades de los modelos basados en *transformers* investigar más a fondo la utilidad de estos y las aplicaciones que tiene para así comprobar sus buenas capacidades.

Palabras clave: análisis sentimental, aprendizaje profundo, *transformers*, redes sociales, medios periodísticos



Tema FCI

FACULTAD DE
CIENCIAS E
INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

| | |
|-------------------|---|
| PARA OPTAR | : Título Profesional de Ingeniero Informático |
| TEMA | : Análisis de sentimiento en información de medios periodísticos y redes sociales mediante redes neuronales recurrentes |
| ÁREA | : Ciencias de la computación |
| ASESOR | : Dr. César Beltrán Castañón |
| ALUMNO(S) | : Gabriel Héland Zárate Calderón - 20150862 |
| FECHA | : Agosto del 2021 |

DESCRIPCIÓN:

El análisis de sentimiento es una subárea de investigación perteneciente al procesamiento de lenguaje natural, la cual está en constante crecimiento, especialmente por tres factores: la generación de grandes volúmenes de información textual, el avance tecnológico en lo que se refiere al almacenamiento y el avance de los algoritmos de inteligencia artificial para el análisis de esta.

Actualmente el desarrollo del análisis de sentimiento en textos en español es una tarea que se desarrolla en empresas dedicadas al monitoreo de medios de información pública, donde este viene realizándose haciendo uso de procedimientos ambiguos y manuales, los cuales consisten en que expertos en este análisis leen y clasifican la polaridad del sentimiento expresada en los textos periodísticos y diversas publicaciones en redes sociales, es decir clasifican si este expresa algo positivo, negativo o es neutral, según su propio criterio basado en su experiencia. Esta tarea manual genera varios problemas: los resultados no son del todo fiables, en vista que siempre existirá el sesgo de cada experto, restándole valor a dicha tarea; la tarea es consume recurso humano y tiempo, debido a la escasa cantidad de herramientas computacionales existentes para realizar este análisis en textos en español.

Con la problemática identificada en el presente proyecto de fin de carrera se propone implementar un modelo algorítmico que desarrolle de forma automática el análisis de sentimiento en notas periodísticas y textos de microblogs de redes sociales en español.

Para ello se implementaran dos modelos algorítmicos que harán uso de redes neuronales recurrentes basadas en modelos *transformers*, uno entrenado con notas

periodísticas y otro con *microblogs* extraídos de Twitter, esto debido a las diferencias entre estos dos tipos de textos. Adicionalmente, el dominio estará delimitado a temas políticos peruanos sucedidos en el año 2020. Los modelos serán entrenados con conjuntos de datos previamente etiquetados por personal experto según la polaridad de textos que estos expresan, siendo positivo, negativo o neutral, por ello finalmente este podrá clasificar nuevos textos según dichas polaridades.

Para probar el funcionamiento correcto del modelo se desarrollará una aplicación web para pruebas del usuario. Esta interfaz le permitirá al usuario clasificar nuevos textos que pueden ser tanto artículos periodísticos como *microblogs* publicados en redes sociales, esto lo podrá hacer de manera individual, es decir ingresar un texto y clasificarlo, o de manera masiva, donde podrá subir un archivo CSV con un formato específico y este clasificará todos los textos ingresados y retornará dicho archivo con la clasificación realizada.

Finalmente, para probar tanto el funcionamiento del modelo como de la aplicación web se trabajará con un conjunto de 10000 publicaciones de Twitter y un conjunto de 3323 artículos periodísticos los cuales serán clasificados haciendo uso de la aplicación y por ende del modelo realizado.

OBJETIVO GENERAL:

Implementar un modelo algorítmico de análisis de sentimiento en notas periodísticas y *microblogs* de redes sociales en español basado en redes neuronales recurrentes.

OBJETIVOS ESPECÍFICOS:

El presente trabajo de fin de carrera cuenta con tres objetivos específicos para así poder cumplir el objetivo general, los cuales son:

1. Implementar un modelo que haga uso de redes neuronales recurrentes para realizar un análisis de sentimiento sobre un conjunto de datos en español
2. Automatizar el proceso de determinar la polaridad de los sentimientos expresados de notas periodísticas
3. Estructurar y clasificar un dominio de información periodística para realizar un análisis de sentimiento



Tabla de Contenido

| | | |
|-------------|--|----|
| Capítulo 1. | Generalidades | 1 |
| 1.1 | Problemática..... | 1 |
| 1.1.1 | Descripción..... | 1 |
| 1.1.2 | Problema seleccionado | 5 |
| 1.2 | Objetivos | 5 |
| 1.2.1 | Objetivo general | 5 |
| 1.2.2 | Objetivos específicos..... | 5 |
| 1.2.3 | Resultados esperados | 5 |
| 1.2.4 | Mapeo de objetivos, resultados y verificación..... | 6 |
| 1.3 | Métodos y Procedimientos | 8 |
| Capítulo 2. | Marco Conceptual..... | 13 |
| 2.1 | Introducción | 13 |
| 2.2 | Desarrollo del marco | 13 |
| Capítulo 3. | Estado del Arte | 18 |
| 3.1 | Introducción | 18 |
| 3.2 | Objetivos de revisión..... | 18 |
| 3.3 | Preguntas de revisión | 19 |
| 3.4 | Estrategia de búsqueda | 20 |
| 3.4.1 | Criterios de inclusión/exclusión | 21 |
| 3.5 | Formulario de extracción de datos | 22 |
| 3.6 | Resultados de la revisión..... | 23 |

| | | |
|-------------|---|----|
| 3.6.1 | Modelos de deep learning usados para análisis sentimental de información pública de medios periodísticos y redes sociales | 26 |
| 3.6.2 | Técnicas en las que se utilizan los modelos transformers..... | 30 |
| 3.6.3 | Variaciones de análisis sentimentales que aplican Transformers | 33 |
| 3.7 | Conclusiones | 35 |
| Capítulo 4. | Modelos de análisis sentimental para conjuntos de datos en español | 37 |
| 4.1 | Introducción..... | 37 |
| 4.2 | Resultados Alcanzados..... | 37 |
| 4.2.1 | Arquitectura base de red neuronal..... | 37 |
| 4.2.2 | Arquitectura base entrenada y validada con un conjunto de datos en español | 44 |
| 4.2.3 | Modelo para análisis sentimental considerando arquitectura base..... | 48 |
| 4.2.4 | Modelo para análisis sentimental optimizado..... | 56 |
| Capítulo 5. | Automatización de la definición de sentimientos de información pública de medios periodísticos y redes sociales | 71 |
| 5.1 | Introducción..... | 71 |
| 5.2 | Interfaz web para pruebas del usuario | 71 |
| 5.3 | Discusión..... | 77 |
| Capítulo 6. | Dominio de información periodística estructurado y clasificado | 79 |
| 6.1 | Introducción..... | 79 |
| 6.2 | Resultados Alcanzados..... | 79 |
| 6.2.1 | Base de datos textual de artículos periodísticos | 79 |
| 6.2.2 | Base de datos textual de tweets | 80 |
| 6.2.3 | Base de datos textual estructurada y clasificada de la información recolectada..... | 81 |

| | |
|---|----|
| 6.3 Discusión..... | 85 |
| Capítulo 7. Conclusiones y trabajos futuros | 87 |
| 7.1 Conclusiones | 87 |
| 7.2 Trabajos futuros..... | 90 |



Índice de Figuras

| | |
|---|----|
| Figura 1. Arquitectura del modelo Transformer (Vaswani et al., 2017) | 28 |
| Figura 2. Distribución de las métricas de evaluación utilizadas | 29 |
| Figura 3. Diagrama de BERT (COURSERA, 2020)..... | 39 |
| Figura 4. Gráfica de resultados de experimentación en el preprocesamiento para los modelos | 51 |
| Figura 5. Conjunto de datos de artículos periodísticos | 53 |
| Figura 6. Gráfica de resultados de experimentación en los optimizadores de los modelos de publicaciones de redes sociales | 59 |
| Figura 6. Gráfica de resultados de experimentación en los optimizadores de los modelos para artículos periodísticos..... | 62 |
| Figura 7. Interfaz web para pruebas de usuario | 75 |
| Figura 8. Prueba de funcionamiento de los modelos dentro de la interfaz web | 76 |
| Figura 9. Base de datos textual de artículos periodísticos estructurada | 82 |
| Figura 10. Base de datos textual de tweets estructurada | 83 |
| Figura 11. Archivo resultante del análisis de sentimiento de artículos periodísticos..... | 84 |
| Figura 12. Archivo resultante del análisis de sentimiento de tweets..... | 84 |

Índice de Tablas

| | |
|---|----|
| Tabla 1. Mapeo de Objetivo 1..... | 6 |
| Tabla 2. Mapeo de Objetivo 2..... | 7 |
| Tabla 3. Mapeo de Objetivo 3..... | 8 |
| Tabla 4. Tabla de herramientas y métodos | 9 |
| Tabla 5. Criterios PICOC..... | 19 |
| Tabla 6. Resultados de búsqueda..... | 20 |
| Tabla 7. Formulario de extracción de datos..... | 22 |
| Tabla 8. Listado de estudios primarios utilizados..... | 23 |
| Tabla 9. Resultados de distintos enfoques de análisis sentimental | 29 |
| Tabla 10. Objetivos de uso de modelos basados en transformers..... | 32 |
| Tabla 11. Distribución del dataset SMILE luego del preprocesamiento y la partición..... | 40 |
| Tabla 12. Resultados de entrenamiento del modelo base..... | 42 |
| Tabla 13. Resultados de evaluación del modelo base | 43 |
| Tabla 14. Distribución del dataset TASS 2019 luego del preprocesamiento y la partición | 45 |
| Tabla 15. Resultados de entrenamiento del modelo base en español..... | 46 |
| Tabla 16. Resultados de evaluación del modelo base | 47 |
| Tabla 17. Resultados de experimentación para la selección del modelo | 50 |
| Tabla 18. Resultados de evaluación del Modelo Español para tweets..... | 51 |
| Tabla 19. Distribución del dataset de artículos periodísticos luego del preprocesamiento y la partición | 53 |
| Tabla 20. Resultados de entrenamiento del modelo en español para artículos periodísticos | 54 |

| | |
|---|----|
| Tabla 21. Resultados de evaluación del modelo en español para artículos periodísticos..... | 55 |
| Tabla 22. Resultados de experimentación para la optimización del modelo para publicaciones de redes sociales..... | 58 |
| Tabla 22. Resultados de experimentación para la optimización del modelo para artículos periodísticos..... | 60 |
| Tabla 23. Resultados de entrenamiento del modelo en español optimizado para tweets | 63 |
| Tabla 24. Resultados de evaluación del modelo en español optimizado para tweets | 64 |
| Tabla 25. Resultados de entrenamiento del modelo en español optimizado para artículos periodísticos | 66 |
| Tabla 26. Resultados de evaluación del del modelo en español optimizado para artículos periodísticos..... | 67 |



Capítulo 1. Generalidades

1.1 Problemática

1.1.1 Descripción.

El uso de medios digitales y redes sociales va creciendo exponencialmente con el avance progresivo que va teniendo la tecnología, es por ello por lo que en la actualidad existen múltiples medios periodísticos que de manera pública comparten su información en plataformas propias o en las redes sociales (De Ita, 2019). Por consecuencia hay grandes volúmenes de información textual, imágenes, sonido y video que pueden ser analizados (Jiang et al., 2017).

Esta disponibilidad de información permite que actividades como el monitoreo de medios dispongan ahora de mayor insumo para el análisis de información. Esta actividad ha tomado especial interés en nuestros días, tanto para las empresas privadas, como para los gobiernos, en vista que es de interés de ellos conocer qué es lo que se comenta y dice dentro de los medios sociales (Bekkers et.al., 2013). En el monitoreo de medios se pueden realizar distintos tipos de análisis (Zhang and Vos, 2014), siendo uno de especial importancia, la extracción de los sentimientos expresados en los textos el cual es denominado análisis sentimental o de sentimiento. El análisis de sentimiento es una tarea muy popular en el procesamiento de textos que consiste en extraer polaridades de sentimiento expresadas de manera implícita o explícita en textos (J. Tao and Fang, 2020).

En la literatura se han reportado diferentes procedimientos para realizar dicha extracción de sentimiento (Batinca and Treleaven, 2015; Younis,2015), sin embargo, no generan una alta confiabilidad en los resultados obtenidos dado que se deben evadir

diferentes desafíos, dada la dinámica de los eventos (Ebrahimi et.al., 2017). Para el trabajo de análisis y extracción de sentimiento medios sociales, todavía es una tarea semi-automática, siendo que parte del procedimiento se apoyan en herramientas (scrapping, reconocimiento de entidades, transcripciones, entre otros), pero la tarea de definición de la polaridad o sentimiento de un texto se realiza de manera manual, por lo tanto, se genera una dependencia con respecto al *expertise* de los especialistas que realizan esa tarea, debido a que la interpretación de los textos puede variar de una persona a otra, por ejemplo lo que a alguna persona le parece que denota una sensación de rechazo o desagrado, a otro le puede parecer que expresa una opinión neutral, por ello dichas extracciones dependen del criterio del experto.

El tener tareas cuyos resultados son altamente influenciados por el criterio y *expertise* del especialista, genera sesgo, lo cual puede provocar una pérdida de credibilidad, precisión, y así, no poder garantizar resultados óptimos bajo dichas circunstancias estandarizar (Venegas, 2007). Adicionalmente, por tratarse de una tarea manual, factores adicionales son inevitables de presentarse que pueden perjudicar al análisis, como el cansancio, pérdida de concentración lo que va variando según el estado anímico del mismo, la disposición para realizar dicha tediosa tarea y muchos otros factores que afectan al proceso generando resultados ambiguos y sesgados de un análisis que al ser basado en criterios expertos es difícil de estandarizar (Venegas, 2007).

Pero, así como una de las principales causas de dicha ambigüedad son los encargados de la extracción de datos, también se ve una causa potencial de problemas en la variabilidad de la información expresada en las notas periodísticas o en las publicaciones de redes sociales, por lo tanto, la estructuración de la información vendría a ser un problema causa.

Esto se da debido al crecimiento exponencial de dicha información, cuál está empezando a ser considerada como una instancia de big data en la literatura (Zhang et al., 2015) dado al gran volumen, la alta variedad y velocidad con la que crece de información manejada, por ejemplo, Facebook que es una de las redes sociales más utilizadas a nivel mundial, recibe un ingreso de 500 TB de información nueva de manera diaria (Jiang et al., 2017). Por ello manejar dichos grandes volúmenes de información variable y dinámica de manera manual para su clasificación sentimental se vuelve algo utópico.

Gracias a ello, procesamiento de lenguaje natural (PLN) ha tomado gran importancia en los últimos años, siendo una de las áreas tradicionales de la Inteligencia Artificial (Deng and Liu, 2018; McNamara et. al., 2017). El PLN es una de las ramas de la Inteligencia Artificial enfocada a generar una comunicación entre las personas y las computadoras, donde la intención es que la computadora pueda “entender” las oraciones o textos (Cortez, Vega and Pariona, 2011). Diferentes aplicaciones pueden ser desarrolladas con PLN, como reconocimiento de voz, entendimiento de habla, sistemas de diálogo, análisis sintáctico, entre otros (Deng and Liu, 2018) donde el análisis sentimental es una subárea de investigación importante del mismo que dado su desarrollo especialmente durante la última década lo ha colocado como una mini versión de PLN o como un caso especial del PLN.

Dado que el análisis de sentimiento también toca cada área del PLN, que incluyen el análisis léxico, resolución de correferencias, desambiguación, análisis de discurso, extracción de información y análisis semántico. Por lo tanto, el uso de dichas técnicas computacionales podría ayudar con ello, pero también este arraiga nuevos desafíos. El principal desafío a enfrentar vendría a ser la barrera del lenguaje, esto se da debido a que los modelos computacionales orientados al procesamiento de textos, de manera específica a los

que se usan para el análisis sentimental, están desarrollados para trabajar con idiomas específicos, dadas las variaciones morfológicas, de sintaxis y semánticas que existen entre los múltiples lenguajes existentes. Por ejemplo, un modelo que extrae sentimientos expresados en un conjunto de tweets escritos en inglés no funcionará correctamente con tweets en español y mucho menos con tweets japoneses o chinos. Por ende, para realizar análisis sentimentales de información pública de medios periodísticos y redes sociales en español se requerirán modelos hechos para dicho idioma.

En la literatura no se reportan muchos trabajos relacionados al análisis sentimental de conjuntos de datos en español, lo que vendría a generar una alta imprecisión en la determinación del sentimiento en el idioma español al hacer uso dichos modelos. Además de la barrera del lenguaje, ya teniendo en cuenta modelos que trabajan con otros idiomas, basándose en la literatura utilizada para definir el estado del arte de este trabajo, se encuentra que los modelos más comunes de la actualidad orientados al análisis sentimental se centran en analizar fuentes de datos que claramente expresan opiniones, como lo son reviews de diversos tópicos (Paredes, et.al., 2017), o comentarios en redes sociales, microblogs (Pla and Hurtado, 2018; Tellez, et. al. 2017), siendo que se han dado algunas iniciativas para crear corpus (del Arco, et. al., 2016; Diaz-Galiano, et.al., 2019). Sin embargo, no hay una gran cantidad de modelos orientados a la realización de análisis sentimentales en información proveniente de medios periodísticos.

En la sección de anexos de la presente tesis, en el Anexo A se puede encontrar un árbol de problemas que resume la descripción de la problemática a abordar.

1.1.2 Problema seleccionado.

Dado el contexto explicado, se verifica la existencia de procedimientos ambiguos para la realización de análisis de sentimiento de información pública de medios periodísticos y redes sociales, al no estar soportados por las herramientas y las técnicas adecuadas generan resultados ambiguos y segados, que generan resultados con poca credibilidad e incluso imprecisos, quitándole valor y desaprovechando el potencial que tiene dicho análisis

1.2 Objetivos.

1.2.1 Objetivo general.

Implementar un modelo algorítmico de análisis de sentimiento en notas periodísticas y *microblogs* de redes sociales en español basado en redes neuronales recurrentes.

1.2.2 Objetivos específicos.

O1. Implementar un modelo que haga uso de redes neuronales recurrentes para realizar un análisis de sentimiento sobre un conjunto de datos en español

O2. 2. Automatizar el proceso de determinar la polaridad de los sentimientos expresados de notas periodísticas

O3. Estructurar y clasificar un dominio de información periodística para realizar un análisis de sentimiento

1.2.3 Resultados esperados.

O1. Implementar un modelo que haga uso de redes neuronales recurrentes para realizar un análisis de sentimiento sobre un conjunto de datos en español

R1. Arquitectura base de red neuronal

R2. Arquitectura base entrenada y validada con un conjunto de datos en español

R3. Modelo para análisis sentimental considerando arquitectura base

R4. Modelo propuesto optimizado y validado

O2. Automatizar el proceso de determinar la polaridad de los sentimientos expresados de notas periodísticas

R1. Interfaz web para pruebas del usuario

O3. Estructurar y clasificar un dominio de información periodística para realizar un análisis de sentimiento

R1. Base de datos textual de artículos periodísticos

R2. Base de datos textual de tweets

R3. Base de datos textual estructurada y clasificada de la información recolectada

1.2.4 Mapeo de objetivos, resultados y verificación.

Tabla 1. Mapeo de Objetivo 1

| Objetivo: Implementar un modelo que haga uso de redes neuronales recurrentes para realizar un análisis de sentimiento sobre un conjunto de datos en español | | |
|--|---------------------------------------|--|
| Resultado | Medio de verificación | Indicador objetivamente verificable |
| Arquitectura base de red neuronal | -Modelo computacional y código fuente | -Score de precisión mayor al 60%* |

| | | |
|--|---|-----------------------------------|
| | -Reporte de arquitectura de modelo base | |
| Arquitectura base entrenada y validada con un conjunto de datos en español | -Reporte de entrenamiento y validación de arquitectura base | -Score de precisión mayor al 65%* |
| Modelo para análisis sentimental considerando arquitectura base | -Modelo computacional y código fuente -Reporte de modelo de análisis sentimental | -Score de precisión mayor al 70%* |
| Modelo propuesto optimizado y validado | -Reporte de optimización y validación del modelo | -Score de precisión mayor al 75%* |

**Los valores que se exigen como IOV's para la precisión de los modelos fue estimada en base los modelos similares encontrados en la investigación del estado del arte*

Tabla 2. Mapeo de Objetivo 2

| | | |
|--|---|--|
| Objetivo: Automatizar el proceso de determinar la polaridad de los sentimientos expresados de notas periodísticas | | |
| Resultado | Medio de verificación | Indicador objetivamente verificable |
| Interfaz web para pruebas del usuario | -Interfaz desplegada -Software y código fuente | -Pruebas unitarias al 100% |

Tabla 3. Mapeo de Objetivo 3

| Objetivo: Estructurar y clasificar un dominio de información periodística para realizar un análisis de sentimiento | | |
|---|--|--|
| Resultado | Medio de verificación | Indicador objetivamente verificable |
| Base de datos textual de artículos periodísticos | -Archivo de texto que contiene al menos 100 artículos periodísticos referidos a política del Perú del 2020, extraídos de los diarios El Comercio y Gestión | -Cantidad mínima de 100 artículos periodísticos referidos a política del Perú del 2020 |
| Base de datos textual de tweets | -Archivo de texto que contiene al menos 1 000 comentarios de microblogs referidos a política del Perú del 2020 | -Cantidad mínima de 1 000 comentarios de microblogs referidos a política del Perú del 2020 |
| Base de datos textual estructurada y clasificada de la información recolectada | -Archivo de texto con la información estructurada y clasificada | -Porcentaje de artículos periodísticos y comentarios de microblogs clasificados al 100% |

1.3 Métodos y Procedimientos.

Para desarrollar el presente trabajo de tesis y cumpliendo los objetivos específicos planteados, y por ende el objetivo general de un modelo que haga uso de redes neuronales recurrentes para realizar un análisis de sentimiento sobre un conjunto de datos en español, y con ello poder resolver la problemática planteada se mostrará una tabla que detalle las herramientas y métodos a usar por cada resultado esperado definido.

Tabla 4. Tabla de herramientas y métodos

| Resultado | Actividad para obtención de resultado | Herramientas y métodos |
|--|---|---|
| Arquitectura base de red neuronal | Selección de arquitectura base de red neuronal | Python Pytorch Jupyter Notebook Git |
| Arquitectura base entrenada y validada con un conjunto de datos en español | Entrenamiento y validación de arquitectura base con un conjunto de datos en español | Python Pytorch Jupyter Notebook Git |
| Modelo para análisis sentimental considerando arquitectura base | Implementación de modelo para análisis sentimental considerando arquitectura base | Python Pytorch Jupyter Notebook Git |
| Modelo propuesto optimizado y validado | Optimización y validación del modelo propuesto | Python Pytorch Jupyter Notebook Git |
| Interfaz web para pruebas del usuario | Implementación de interfaz web para pruebas del usuario | Python Flask Javascript React Git |

| | | |
|--|---|------------------------|
| Base de datos textual de artículos periodísticos | Recolección de artículos periodísticos y limpieza de la información | Python Archivos CSV |
| Base de datos textual de tweets | Recolección de tweets y limpieza de la información | Python Archivos CSV |
| Base de datos textual estructurada y clasificada, de la información recolectada con participación de experto | Estructuración del conjunto de datos recolectado con participación de experto | Python Pandas |

A continuación, se presentará una definición detallada de las herramientas que serán utilizadas para obtener los resultados esperados presentados en este trabajo de investigación.

- **Jupyter Notebook:** Es una aplicación web de código abierto que permite la creación de documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo, que comúnmente es usada para la limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático (Project Jupyter, n.d.). En este proyecto será utilizado para las tareas de la estructuración del conjunto de datos y para la implementación del modelo de análisis sentimental.
- **Git:** Es un sistema de control de versiones distribuido y gratuito de código abierto, que es fácil de aprender y tiene una huella pequeña con rendimiento increíblemente

rápido (Git, n.d.). Será utilizado para gestionar las versiones del modelo a implementar y de la aplicación web que trabajará con el mismo.

- **Python:** Es un lenguaje de programación interpretado de programación multiparadigma, ya que soporta programación orientada a objetos, imperativa y funcional (Python, n.d.), que además potencia los principales aspectos de ciclo de vida de aprendizaje de máquina, incluida la anotación de datos, la investigación y la experimentación, y el despliegue del modelo a producción (Nimshi and Konam, 2020). Este lenguaje de programación será la principal herramienta bajo la cual se va a implementar el modelo de análisis sentimental.
- **Pandas:** Es una herramienta de análisis y manipulación de datos de código abierto rápida, potente, flexible y fácil de usar, construido sobre el lenguaje de programación Python (pandas, n.d.). Para este proyecto se va a utilizar esta herramienta para limpiar y estructurar la información recolectada de medios periodísticos y redes sociales para posteriormente ser usada como conjunto de datos de entrada en el modelo.
- **Pytorch:** Es un marco de trabajo de aprendizaje automático de código abierto que acelera el camino desde la creación de prototipos de investigación hasta el despliegue de producción (PyTorch, n.d.). Esta herramienta será bajo la cual se implementará la arquitectura del modelo, donde se usarán redes neuronales basadas en Transformers.
- **Flask:** Es un marco de trabajo que sirve para desarrollar aplicaciones web en Python, ya que soporta implementaciones tanto de *backend* como *frontend* (Flask, n.d.). Esta herramienta será utilizada para desarrollar el *backend* de la aplicación web desarrollada en el proyecto.

- **Javascript:** Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos orientados a objetos, imperativos y declarativos (Mozilla, n.d.). Esta herramienta será el lenguaje base a utilizar para la interfaz web que se usará para las pruebas de usuario.
- **React:** Es una biblioteca de JavaScript para construir interfaces de usuario que es declarativo y está basado en componentes (React, n.d.). Será utilizada para el desarrollo del *frontend* de la aplicación web desarrollada en el proyecto.



Capítulo 2. Marco Conceptual

2.1 Introducción.

El análisis de sentimiento es comúnmente visto como una subárea del PLN, sin embargo, su desarrollo especialmente durante la última década lo ha colocado como una mini versión de PLN o como un caso especial del PLN, la razón es que el análisis de sentimiento también toca cada área del PLN, como el análisis léxico, resolución de correferencias, desambiguación, análisis de discurso, extracción de información y análisis semántico. En general, el análisis de sentimiento es un problema de análisis semántico, pero no se requiere entender completamente toda la sentencia o documento; sólo es necesario entender algunos aspectos de ello, o poner atención a elementos que manifiesten el sentimiento dentro de un texto.

En el presente trabajo se abordará el análisis de sentimiento dentro de notas periodísticas, para lo cual en este capítulo se presentan las principales definiciones para entender cómo funciona un análisis sentimental basado en datos de información pública de medios periodísticos y redes sociales, es por ello por lo que se explicará de manera concisa desde lo más básico del procesamiento del lenguaje natural para luego ir entrando a temas un poco más avanzados de dicho rubro aplicados a los análisis de sentimiento, como vendrían a ser los mecanismos de atención.

2.2 Desarrollo del marco.

2.2.1 Análisis de información pública de medios periodísticos y redes sociales.

Con el crecimiento exponencial del uso de medios digitales y redes sociales, se va generando un aumento masivo de información en todas estas plataformas, generando mucha

información que debidamente procesada puede resultar valiosa para los clientes, los negocios y los prestadores de múltiples servicios. Esto se debe a la gran cantidad de información implícita o explícita que cada post, comentario, noticia, foro o cualquier tipo de contenido multimedia digital contiene, ya que estos pueden ser utilizados para expresar opiniones, con lo cual se pueden analizar tendencias y esto se puede aplicar a múltiples negocios.

Zhang menciona que la información de las redes sociales se considera como una instancia de *big data* en la literatura (Zhang *et al.*, 2015). Esto se argumenta en las siguientes razones:

- Volumen: el número promedio de *microblogs* que reportan un evento noticioso es de al menos 100,000
- Variedad: cada *microblog* está compuesto de distinto material (tales como palabras o emoticones) que expresan múltiple información
- Velocidad: la información manejada en las redes sociales es altamente dinámica. Por ejemplo, se menciona que aproximadamente cada día ingresan 500TB de información nueva a Facebook
- Valor: la información en bruta “escondida” en las publicaciones de las redes sociales es el terreno de prueba perfecto para los investigadores abocados al área de *big data* y también es una poderosa herramienta para las corporaciones y gobiernos en su toma de decisiones o definición de estrategias globales.

(Jiang *et al.*, 2017)

2.2.2 Procesamiento del Lenguaje Natural (PLN).

El procesamiento del lenguaje natural es una de las ramas de la Inteligencia Artificial que consiste en la utilización del lenguaje natural para que haya una comunicación entre las

personas y las computadoras, donde la intención es que la computadora pueda “entender” las oraciones o textos que se le sean proporcionados a través del uso de diversas técnicas de procesamiento (Cortez, Vega and Pariona, 2011).

Este cuenta con las siguientes áreas:

- **Análisis léxico:** consiste en el análisis de las gramáticas lexicalizadas, las cuales ponen más énfasis en la estructura del léxico, disminuyendo de esta manera la carga sobre las reglas de estructura de frase pura.
- **Resolución de correferencias:** consiste en poder resolver las correferencias hechas en los textos con respecto a menciones o expresiones de referencias hechas a alguna entidad, por ejemplo:

“El profesor llegó tarde al salón. Este hombre siempre es muy impuntual, lo cual da una mala imagen de él.”

En el texto mostrado, las palabras subrayadas se refieren a la misma persona, pero solo en la primera vez se hace de manera directa, luego son correferencias hechas a ella.

- **Desambiguación:** como su nombre lo dice, consisten en la desambiguación sintáctica de un texto, donde se sea capaz de elegir un solo análisis gramatical correcto dentro de una multitud de posibles análisis, para entender mejor ello, se explicará que una ambigüedad estructural ocurre cuando la gramática puede asignar más de un análisis gramatical a una oración.
- **Análisis de discurso:** consiste en analizar un grupo estructurado coherente de oraciones, el cual es denominado como discurso, teniendo en cuenta la “herencia” de

coherencia entre todas las oraciones que componen el discurso, para así analizarlas todas en conjunto, por ejemplo, se puede considerar el analizar un capítulo entero de una novela.

- **Extracción de información:** es el proceso de convertir la información no estructurada contenida en textos en conjuntos de datos estructurados, extrayendo diferentes tipos de contenido semántico de estos, por ejemplo, para llenar una base de datos relacional para su posterior procesamiento.
- **Análisis semántico:** consiste en el análisis de las palabras en base a su sentido, que será una representación discreta del significado de esta. Por ejemplo, esto se puede aplicar con palabras que tengan doble sentido, como lo es ratón, que puede ser o el animal, o el dispositivo que se usa para controlar el cursor en el computador, donde dependerá del contexto en el que se use.

(Jurafsky and Martin, 2002)

Si bien hay múltiples ejemplos de la aplicación del procesamiento del lenguaje natural, en el siguiente inciso se explica cómo este es utilizado para abordar detección de sentimientos o emociones expresadas en textos, el cuál es denominado como análisis de sentimiento y es considerado como una subárea del PLN, sin embargo, su desarrollo especialmente durante la última década empieza a ser colocado como una mini versión de PLN o como un caso especial del PLN, por el hecho de que el análisis de sentimiento también desarrolla cada una de las tareas típicas, de forma abreviada, del PLN.

2.2.3 Análisis Sentimental.

El análisis sentimental es el proceso de analizar, procesar, resumir e inferir texto subjetivo con polaridad de sentimiento, que es una tarea vital en el procesamiento del lenguaje natural (PLN) y también se conoce como minería de opinión (Zhang and Liu, 2016).

Este es una tarea común de la clasificación de textos, que consiste en la extracción del sentimiento, ya sea positivo o negativo, expresado por el autor con respecto a algún tema. La versión más simple de dicho análisis consiste en una clasificación binaria de comentarios u opiniones donde se las palabras utilizadas expresan claramente una opinión bien marcada, como es el caso de *reviews* de películas o restaurantes (Jurafsky and Martin, 2002).

A continuación, se muestra un ejemplo básico de la clasificación binaria de sentimientos:

Positivo: ... la trama fue muy **buena, excelente** final.

Negativo: ... la pizza estuvo **horrible** y el precio era **ridículamente** alto.

El poder descubrir y extraer el sentimiento de una noticia o un simple posteo o comentario de alguna red social puede ser utilizado en una variedad de configuraciones. Por ejemplo:

- Monitoreo de la opinión pública: esto puede ayudar a la toma de decisiones al momento de ayudar a comprender los intereses del público con respecto a algún evento concreta
- Monitoreo de retroalimentación de productos: servirá para tener una información real de las opiniones públicas con respecto a algún producto centrándose en las emociones descritas por el público con respecto a algún producto en concreto, para así evaluar resultados y tomar decisiones con respecto a futuros diseños.

(Jiang *et al.*, 2017)

Capítulo 3. Estado del Arte

3.1 Introducción

El análisis de sentimiento es un área de investigación importante en el procesamiento de lenguaje natural, la cual está en constante crecimiento especialmente por la generación de grandes volúmenes de información textual, y el avance tecnológico en lo que se refiere al almacenamiento y los algoritmos inteligentes para el análisis de esta. Además, cada vez va tomando más fuerza su uso en diferentes aplicaciones computacionales dado el crecimiento exponencial del uso de medios digitales y redes sociales, las cuales, gracias a la información debidamente procesada, pueden ser muy valiosas para los negocios. Para realizar dicha tarea existen muchas técnicas y distintos modelos a usar, especialmente durante los últimos años, las estrategias basadas en aprendizaje profundo (*deep learning*) se han mostrados muy adecuadas, siendo que el último año uno de los que va siendo reconocido cada vez más es el uso de modelos basados en Transformers, dadas sus buenas capacidades en el procesamiento de lenguaje natural.

Por ello a continuación se realizará una revisión sistemática del estado del arte del análisis sentimental y del uso de modelos de aprendizaje profundo basados en Transformers.

3.2 Objetivos de revisión.

Esta revisión sistemática tendrá como principales objetivos realizar una comparativa entre los modelos de aprendizaje profundo que son utilizados para realizar análisis sentimental de información pública de medios periodísticos y redes sociales, luego se enfocará la revisión hacia los modelos basados en *transformers* identificando en qué tipo de técnicas se aplican estos, para finalmente centrarse en qué tipos de análisis sentimental se aplicaron dichos modelos.

Para estructurar los elementos de los objetivos de la revisión se mostrará una tabla que hace uso de los criterios PICOC.

Tabla 5. Criterios PICOC

| | |
|--------------|---|
| Population | -Análisis sentimental de textos que use cualquier técnica de aprendizaje profundo -Modelos basados en <i>transformers</i> aplicados a cualquier técnica de aprendizaje |
| Intervention | Aplicación de aprendizaje profundo para procesamiento del lenguaje natural |
| Comparison | Entre modelos de aprendizaje profundo utilizados para análisis sentimental |
| Outcome | Ventajas y desventajas del uso de modelos <i>transformers</i> para el análisis de sentimiento |
| Context | Estudios de investigación que pertenecen al área de las ciencias de la computación orientadas al uso de aprendizaje profundo para procesamiento del lenguaje natural |

3.3 Preguntas de revisión.

- P1. ¿Qué modelos de deep learning son usados para el análisis sentimental de información pública de medios periodísticos y redes sociales?
 - P1.1 ¿Qué métricas de evaluación se utilizaron y qué valores tuvieron las mismas?
 - P1.2 ¿De qué manera son aplicados los resultados obtenidos en cuanto a casos de estudio reales?
- P2. ¿Para qué tipo de técnicas se aplicaron los modelos transformers en el procesamiento de lenguaje natural?

- P2.1 ¿Con qué objetivo fueron diseñados dichos modelos?
- P3. ¿Qué variaciones de análisis sentimental utilizaron los modelos basados en Transformers?
 - P3.1 ¿Qué tipos de conjuntos de datos se usaron y qué embeddings se utilizaron?
 - P3.2 ¿En qué categorías se clasificaron los textos?

3.4 Estrategia de búsqueda.

Se utilizarán dos motores de búsqueda, IEEE Xplore y Scopus, para los cuales se definió una cadena de búsqueda por pregunta:

Tabla 6. Resultados de búsqueda

| Motor de Búsqueda | Cadenas generales básicas de búsquedas | Cantidad de resultados | Preguntas relacionadas |
|-------------------|--|------------------------|------------------------|
| IEEE Xplore | "sentiment* analysis" AND "Natural Language Processing" AND "deep learning" and (news OR "social media") | 3 | 1 |
| IEEE Xplore | deep learning AND Natural Language Processing and transformer* and attention | 3 | 2 |
| IEEE Xplore | "Sentiment* analysis" AND "Transformer*" | 6 | 3 |
| Scopus | TITLE-ABS-KEY("sentiment* analysis" AND "Natural Language Processing" AND "deep learning" and (news OR "social media")) AND (LIMIT-TO (openaccess,1)) AND (LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO (PUBYEAR,2019)) | 7 | 1 |

| | | | |
|--------|---|---|---|
| | OR LIMIT-TO (PUBYEAR,2018) OR LIMIT-TO (PUBYEAR,2017)) | | |
| Scopus | TITLE-ABS-KEY(deep learning AND Natural Language Processing and transformer* and attention) AND (LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO (PUBYEAR,2019) OR LIMIT-TO (PUBYEAR,2018) OR LIMIT-TO (PUBYEAR,2017)) AND (LIMIT-TO (openaccess,1)) | 9 | 2 |
| Scopus | TITLE-ABS-KEY("Sentiment* analysis" AND "Transformer*") AND (LIMIT-TO (openaccess,1)) AND (LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO (PUBYEAR,2019) OR LIMIT-TO (PUBYEAR,2018) OR LIMIT-TO (PUBYEAR,2017)) | 4 | 3 |

3.4.1 Criterios de inclusión/exclusión.

Se incluirán los estudios que cumplan con los siguientes criterios:

- El estudio presenta resultados en sus métricas mayores a 65%
- El estudio está escrito en inglés o español.
- El estudio tiene resultados experimentales replicables

Se excluirán los estudios que cumplan con los siguientes criterios:

- El estudio utiliza técnicas de aprendizaje automático tradicionales que no realizan una extracción automática de las características
- El estudio tiene una antigüedad mayor a 4 años (2017), puesto que la idea es comparar los modelos con los que están hagan uso de redes neuronales recurrentes

transformers, y el primer artículo publicado con dicho enfoque orientado al uso de capas de atención para NLP fue publicado en dicho año.

- Los estudios no son de acceso abierto

3.5 Formulario de extracción de datos.

Se diseñó un formulario genérico que pueda ser aplicado a las tres preguntas planteadas:

Tabla 7. Formulario de extracción de datos

| Campo | Descripción | Ejemplo | Pregunta |
|----------------------|----------------------------------|---|----------|
| Id | | E01 | General |
| Fecha de extracción | | 24/07/2020 | General |
| Autores | | <u>Zihao Fu</u> | General |
| Título | | Transformer Based Memory Network for Sentiment Analysis of Web Comments | General |
| Año de Publicación | | 2019 | General |
| Tipo de Bibliografía | | Revista, paper, congreso, etc. | General |
| Motor de búsqueda | | IEEE Xplore | General |
| Modelo | Qué modelo se utilizó | Sequence To Sequence, Transformer | P1 |
| Técnica | Qué técnica de NLP se desarrolla | Sentiment Analysis, Named Entity Recognition | P2 |

| | | | |
|--------------------------------------|--|--|------------|
| Corpus | Qué tipo de información se usará | Noticias, Web Comments, Tweets | P3 |
| Lenguaje del conjunto de datos | En qué lenguaje está la del conjunto de datos a usar | Español, inglés | P1, P2, P3 |
| Embeddings | Qué embeddings se usaron para vectorizar las palabras | word2vec | P3 |
| Problemática abordada | Qué problemas se abordaron en el procesamiento de NLP | Análisis de aceptación de un producto, determinar la opinión pública de un político | P2 |
| Casos de estudio reales y aplicación | En qué casos de estudio reales se aplicaron | Identificación de cyberbullying | P1, P3 |
| Métrica de evaluación y resultado | Qué métrica se utilizó para evaluar el modelo y el valor de esta | Score: 71.82% | P1 |
| Target | En qué categorías era clasificada la información finalmente | Por polaridad: "positivo", "negativo", "neutral" Por sentimiento: "enfado", "tristeza", o "felicidad" | P3 |

3.6 Resultados de la revisión.

Para realizar la revisión sistemática se utilizaron los siguientes estudios primarios:

Tabla 8. Listado de estudios primarios utilizados

| ID | Título | Autores | Año de Publicación | Tipo de Bibliografía | Motor de búsqueda |
|----|--------|---------|--------------------|----------------------|-------------------|
|----|--------|---------|--------------------|----------------------|-------------------|

| | | | | | |
|-----|--|---|------|------------------|----------------------|
| E01 | A Multi-Layer Dual Attention Deep Learning Model with Refined Word Embeddings for Aspect-Based Sentiment Analysis | S. Rida-E-Fatima; A. Javed; A. Banjar; A. Irtaza; H. Dawood; H. Dawood; A. Alamri | 2019 | Artículo | IEEE Xplore |
| E02 | Toward multi-label sentiment analysis: a transfer learning-based approach | Tao, J., Fang, X. | 2020 | Artículo | Scopus |
| E03 | ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing | Ghorbani, M., Bahaghighat, M., Xin, Q., Özen, F. | 2020 | Artículo | Scopus |
| E04 | Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics | Ait Hammou, B., Ait Lahcen, A., Mouline, S. | 2020 | Artículo | Scopus |
| E05 | Classification of sentiments on online products using deep learning model – RNN | Lakshmidēvi, N., Vamsikrishna, M., Nayak, S.S. | 2019 | Artículo | Scopus |
| E06 | A combined deep learning model for Persian Sentiment Analysis | Nezhad, Z.B., Deihimi, M.A. | 2019 | Artículo | Scopus |
| E07 | Senzi: A sentiment analysis lexicon for the latinised Arabic (Arabizi) | Tobaili, T., Fernandez, M., Alani, H., Sharafeddine, S., Hajj, H., Glavaš, G. | 2019 | Conference Paper | Scopus |
| E08 | A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification | Zheng, J., Zheng, L. | 2019 | Artículo | IEEE Xplore / Scopus |
| E09 | Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network | Lien Minh, D., Sadeghi-Niaraki, A., Huy, H.D., Min, K., Moon, H. | 2018 | Artículo | IEEE Xplore / Scopus |
| E10 | HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention | Bao, Q., Ni, L., Liu, J. | 2020 | Conference Paper | Scopus |
| E11 | A Fusion Model-Based Label Embedding and Self-Interaction Attention for Text Classification | Dong, Y., Liu, P., Zhu, Z., Wang, Q., Zhang, Q. | 2020 | Artículo | IEEE Xplore / Scopus |

| | | | | | |
|-----|--|---|------|------------------|----------------------|
| E12 | Emotional conversation generation based on a Bayesian deep neural network | Sun, X., Li, J., Wei, X., Li, C., Tao, J. | 2019 | Artículo | Scopus |
| E13 | Concept to code: Deep learning for multitask recommendation | Sonie, O. | 2019 | Conference Paper | Scopus |
| E14 | Self-attention networks for intent detection | Yolchuyeva, S., Németh, G., Gyires-Tóth, B. | 2019 | Conference Paper | Scopus |
| E15 | Transformer-Based Neural Network for Answer Selection in Question Answering | Shao, T., Guo, Y., Chen, H., Hao, Z. | 2019 | Artículo | IEEE Xplore / Scopus |
| E16 | Dependency-based self-attention for transformer NMT | Deguchi, H., Tamura, A., Ninomiya, T. | 2019 | Conference Paper | Scopus |
| E17 | Understanding neural machine translation by simplification: The Case of Encoder-free Models | Tang, G., Sennrich, R., Nivre, J. | 2019 | Conference Paper | Scopus |
| E18 | R-Trans: RNN Transformer Network for Chinese Machine Reading Comprehension | S. Liu, S. Zhang, X. Zhang and H. Wang, | 2019 | Artículo | IEEE Xplore |
| E19 | Transformer Based Memory Network for Sentiment Analysis of Web Comments | M. Jiang; J. Wu; X. Shi; M. Zhang | 2019 | Artículo | IEEE Xplore / Scopus |
| E20 | Target-Dependent Sentiment Classification With BERT | Z. Gao; A. Feng; X. Song; X. Wu | 2019 | Artículo | IEEE Xplore / Scopus |
| E21 | Interactive Multi-Head Attention Networks for Aspect-Level Sentiment Classification | Q. Zhang; R. Lu; Q. Wang; Z. Zhu; P. Liu | 2019 | Artículo | IEEE Xplore |
| E22 | R-Transformer Network Based on Position and Self-Attention Mechanism for Aspect-Level Sentiment Classification | Z. Zhou; F. Liu; Q. Wang | 2019 | Artículo | IEEE Xplore |
| E23 | Capsule Network With Identifying Transferable Knowledge for Cross-Domain Sentiment Classification | H. Yin; P. Liu; Z. Zhu; W. Li; Q. Wang | 2019 | Artículo | IEEE Xplore |
| E24 | Sentiment Classification Based on Part-of-Speech and Self-Attention | K. Cheng; Y. Yue; Z. Song | 2020 | Artículo | IEEE Xplore |

| | | | | | |
|-----|--|-------------------------------|------|----------|--------|
| | Mechanism | | | | |
| E25 | Multi-source data fusion for aspect-level sentiment classification | Chen, F., Yuan, Z., Huang, Y. | 2020 | Artículo | Scopus |
| E26 | A Multi-Attention Network for aspect-level sentiment analysis | Zhang, Q., Lu, R. | 2019 | Artículo | Scopus |

Para resolver las preguntas de revisión se llenó el formulario de extracción con la información pertinente de cada estudio y se procedió a adjuntar dichos resultados al final del documento en el Anexo B.

3.6.1 Modelos de deep learning usados para análisis sentimental de información pública de medios periodísticos y redes sociales.

¿Qué modelos de deep learning son usados para el análisis sentimental de información pública de medios periodísticos y redes sociales?

Los modelos utilizados para realizar análisis sentimentales utilizando información pública de medios periodísticos y redes sociales que se revisaron son combinaciones o modelos mejorados de los mostrados a continuación:

3.6.1.1 Long short-term memory (LSTM).

La red LSTM es un tipo especial de red neuronal recurrente compuesto por celdas especiales que principalmente es diseñada para aprender dependencias de largo plazo utilizando celdas de memoria para almacenar información (Ait Hammou, Ait Lahcen and Mouline, 2020).

Cada celda en este modelo cuenta con tres compuertas:

- *Forget gate*: controla cuánto del anterior estado a largo plazo se lee al estado actual

- *Input gate*: controla cuánto del input actual es añadido al estado actual
- *Output gate*: decide qué partes del estado a largo plazo deben estar en el siguiente estado oculto

Esta red también presenta una extensión de esta, la cuál es la *Bidirectional Long Short-Term Memory* (BiLSTM), el cual consiste en utilizar dos redes LSTM para así aprovechar al máximo tanto el contexto a futuro como el pasado (Ait Hammou, Ait Lahcen and Mouline, 2020).

3.6.1.2 Gated recurrent unit (GRU).

La red GRU es una variante de la red LSTM diseñada para simplificar la compleja arquitectura de las LSTM reduciendo el número de parámetros a entrenar en cada celda, contando ahora con dos compuertas: *reset gate* y *update gate*, teniendo así un mejor performance que muchos modelos (Ait Hammou, Ait Lahcen and Mouline, 2020).

3.6.1.3 Modelos basados en mecanismos de atención.

La técnica de atención es usada para seleccionar las partes más importantes del input (Rida-E-Fatima *et al.*, 2019). Estos mecanismos le asignan distintos pesos a cada palabra de la oración dependiendo de su importancia en la misma, facilitando la distinción de palabras clave que juegan un rol de mayor importancia al momento de la clasificación (Zheng and Zheng, 2019)

Existen modelos que pretenden explotar las capacidades de dicha técnica haciendo uso de técnicas más sofisticadas del mismo, como es el uso de la atención dual, donde se explota la relación de propagación dual de los términos, como por ejemplo en cuanto a aspectos y a sentimientos (Rida-E-Fatima *et al.*, 2019)

3.6.1.4 Modelos basados en *transformers*.

Es una arquitectura basada en el mecanismo de auto-atención, el cual evita la recurrencia y en su lugar confía completamente del mecanismo de atención para extraer dependencias globales entre entradas y salidas, con lo cual se permite un incremento significativo en lo que es la paralelización al momento de entrenar dicho modelo (Vaswani *et al.*, 2017). La Figura 3.1 muestra la arquitectura del modelo.

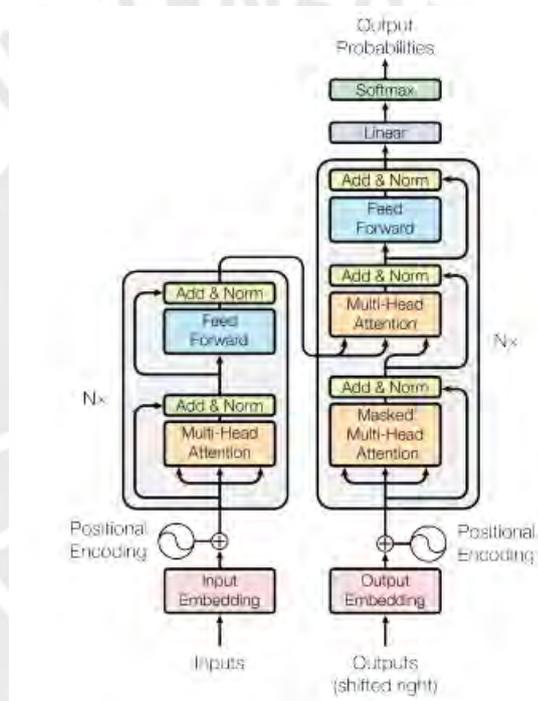


Figura 1. Arquitectura del modelo Transformer (Vaswani *et al.*, 2017)

3.6.1.5 Métricas de evaluación.

¿Qué métricas de evaluación se utilizaron y qué valores tuvieron las mismas?

En los estudios revisados para resolver esta pregunta de revisión se utilizaron tres métricas: *accuracy* (precisión), *F1-Score* y *Loss*, en la Figura 2 se puede ver la distribución del uso de estas métricas:

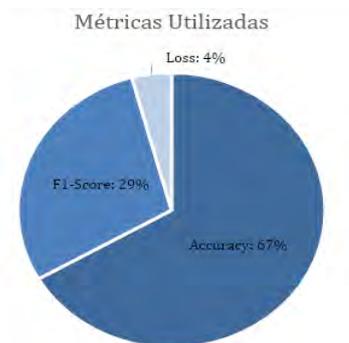


Figura 2. Distribución de las métricas de evaluación utilizadas

Los resultados de las métricas obtenidos no necesariamente implican que un modelo es mejor de otro realizando un análisis sentimental, puesto que hay diversos enfoques de dicho análisis, como son los que están basados en aspectos (ABSA) o en aspectos mejorados (AESA), además de que en algunos casos se usan conjuntos de datos con etiquetas múltiples y de que depende del idioma del conjunto de datos, por ello en la Tabla 9 se muestran los mejores resultados de cada enfoque teniendo en cuenta modelos que trabajan con conjuntos de datos en inglés.

Tabla 9. Resultados de distintos enfoques de análisis sentimental

| Modelo | Enfoque de AS | Métrica y valor |
|---|--|------------------|
| Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model | Análisis sentimental | Accuracy: 96.81% |
| Multi-Attention Network | Análisis sentimental basado en aspectos (ABSA) | Accuracy: 91.22% |
| XLNet (Transfer Learning) | Análisis sentimental basado en aspectos mejorados (AESA) con etiquetas múltiples | Accuracy: 89.96% |

| | | |
|---|---|-----------------|
| Capsule network method with Identifying Transferable Knowledge (CITK) | Análisis sentimental de dominios cruzados | Accuracy: 94.6% |
|---|---|-----------------|

3.6.1.6 Aplicaciones de los modelos.

¿De qué manera son aplicados los resultados obtenidos en cuanto a casos de estudio reales?

Los análisis sentimentales pretenden facilitar la extracción dicho tipo de información importante contenida en noticias y publicaciones de las redes sociales para con ello comprender mejor las opiniones públicas sobre diversos temas y así utilizar esta información según mejor sea el caso. Por ejemplo, se da el caso de un modelo que en base a noticias financieras e información histórica de los precios de las acciones se genera un modelo que predice tendencias de los precios de las acciones a corto plazo (Lien Minh *et al.*, 2018) y también hay un modelo que se utilizó en un sistema inteligente distribuido para análisis de big data social en tiempo real (Ait Hammou, Ait Lahcen and Mouline, 2020).

3.6.2 Técnicas en las que se utilizan los modelos transformers

¿Para qué tipo de técnicas se aplicaron los modelos transformers en el procesamiento de lenguaje natural?

El hecho de que la arquitectura de los modelos basados en Transformers utilice los mecanismos de autoatención para la extracción de dependencias entre palabras resulta muy

útil y esto se ve reflejado en su uso en las siguientes técnicas, dentro de las cuales todas son orientadas al procesamiento de lenguaje natural:

- **Selección de respuestas en sistemas de respuestas de preguntas (QA):** consiste en seleccionar la respuesta más apropiada para una pregunta dentro de un grupo de posibles respuestas (Shao *et al.*, 2019).
- **Generación de textos emocionales:** es una subrama de la generación de textos de lenguaje natural pero enfocado a generar texto natural y coherente con factores emocionales específicos teniendo en cuenta tanto el sentido del texto en sí como del sentimiento (Sun *et al.*, 2019).
- **Detección de intenciones:** esta tarea se enfoca en clasificar la intención del sujeto en el texto y extraer conceptos semánticos como restricciones del lenguaje natural (Yolchuyeva, Németh and Gyires-Tóth, 2019).
- **Machine reading comprehension (MCR):** consiste en darle a la máquina la capacidad de resolver preguntas basadas en el texto proporcionado (Liu *et al.*, 2019).
- **Traducción de textos**
- **Clasificación de textos:** tiene como principal objetivo anotar una secuencia de texto dada con una o varias etiquetas que describen la semántica textual (Dong *et al.*, 2020)
- **Análisis sentimental:** es el proceso de analizar, procesar, resumir e inferir texto subjetivo con polaridad de sentimiento, que es una tarea vital en el procesamiento del lenguaje natural (NPL) y también se conoce como minería de opinión (Zhang and Liu, 2016).

3.6.2.6 Objetivos de uso de modelos Transformers.

¿Con qué objetivo fueron diseñados dichos modelos?

A continuación, se mostrará un listado de los objetivos de los estudios revisados para los cuales se desarrolló cada modelo para las técnicas explicadas:

Tabla 10. Objetivos de uso de modelos basados en transformers

| Técnica | Cita del artículo | Objetivo |
|---|--|--|
| Selección de respuestas en sistemas de respuestas de preguntas (QA) | (Shao <i>et al.</i> , 2019) | Se desea hacer un modelo enfocado en la selección de respuestas, subtarea de los sistemas de QA utilizando modelos basados en Transformers aprovechando que estos se basan solo en la atención y que son buenos para extraer características. |
| Generación de textos emocionales | (Sun <i>et al.</i> , 2019) | Se propone un modelo que genere conversaciones basados en una red neuronal profunda Bayesiana que genere respuestas ricas en emociones, temas claros y diversas, que luego serán divididas en múltiples cláusulas y de ahí pasarán por un modelo basado en <i>transformer</i> que genere cláusulas bidimensionales: granularidad y estructura de las oraciones |
| Detección de intenciones | (Yolchuyeva, Németh and Gyires-Tóth, 2019) | Se desea mostrar mejoras en el uso de un modelo <i>transformers</i> y un codificador de oraciones universal basado en redes de promedio profundo para la tarea de detección de entidades |
| Machine reading comprehension (MCR) | (Liu <i>et al.</i> , 2019) | Se desea aplicar modelos existentes de MRC a documentos chinos mitigando la influencia de la segmentación incorrecta y minar la información secuencial de oraciones completas |
| Traducción de textos | (Deguchi <i>et al.</i> , 2019) | Se desea ampliar LISA (linguistically-informed self-attention) que inicialmente era usado para el codificador de etiquetado semántico de las Transformers para enmascarar información futura de las palabras en el lado del decodificador en la autoatención basada en dependencias |
| Clasificación de textos | (Dong <i>et al.</i> , 2020) | Se propone hacer un modelo basado en mecanismos de atención de auto-interacción y embeddings de etiquetas, teniendo en cuenta la interacción de información entre todas las oraciones de un texto para que no haya pérdida de semántica |
| Análisis sentimental | (Zhang & Lu, 2019) | Se desea hacer un modelo de atención múltiple que no falle al extraer representaciones de texto de manera eficiente y que logre la interacción entre términos y aspectos |

3.6.3 Variaciones de análisis sentimentales que aplican *transformers*.

¿Qué variaciones de análisis sentimental utilizaron los modelos basados en transformers?

El análisis sentimental consiste en extraer polaridades de sentimiento expresadas de manera implícita o explícita en textos, y por lo tanto los sentimientos expresados muchas veces dependen de otros factores expresados en los textos, por ello se dan variaciones en estos análisis para tratar estos casos y orientarlos a tareas más específicas. Según los estudios revisados, las variaciones encontradas fueron las siguientes:

3.6.3.1 Análisis sentimental basado en aspectos (ABSA).

Esta consiste en extraer características o aspectos específicos de distintos bienes o servicios que están asociados directamente con los sentimientos expresados por los clientes. Existen dos principales métodos que usan para este tipo de extracción: basados en lexicón y basados en aprendizaje de máquina (Jie Tao and Fang, 2020).

3.6.3.2 Clasificación de etiquetas múltiples.

Consiste en clasificar instancias en un subconjunto de clases predefinidas, lo cual le aumenta complejidad a la clasificación, pero su uso en los análisis sentimentales se da por el hecho de que en los textos reales normalmente se discuten múltiples temas. Existen 3 categorías para agrupar estos métodos: métodos basados en la transformación de problemas, métodos basados en la adaptación de algoritmos y métodos basados en conjuntos (J. Tao and Fang, 2020).

3.6.3.3 Análisis sentimental basado en aspectos mejorado (AESAs).

Es una mejora del ABSA que intenta realzar los sentimientos a nivel de documento con aspectos discutidos en todo el documento, esto quiere decir que se enfocará en el contexto expresado en todo el documento, es por ello por lo que esta clasificación, por naturaleza, será de etiquetas múltiples (Jie Tao and Fang, 2020).

3.6.3.4 Análisis sentimental de dominios cruzados.

Como su nombre lo menciona, consiste en utilizar dominios cruzados, es decir, entrenar al modelo con un conjunto de datos perteneciente a un dominio específico y luego aplicarlo a otro. Dentro de este tipo de clasificación, los esfuerzos más importantes son el aprendizaje por correspondencia de estructura (SCL), el cual puede predecir la presencia de características de pivote en los targets del conjunto de datos, y la alineación estructurada de características (SFA), que logra la clasificación cruzada usando palabras independientes del dominio (Yin *et al.*, 2019).

3.6.3.5 Tipo de conjuntos de datos y *embeddings* utilizados.

*¿Qué tipos de conjuntos de datos se usaron y qué *embeddings* se utilizaron?*

Los estudios revisados, al momento de realizar las experimentaciones para probar sus modelos, utilizaron principalmente “reviews” extraídos de distintas fuentes (Amazon, TBS, Yelp, etc.) de diversos tópicos (hoteles, restaurantes, electrodomésticos, etc.), de los cuales la gran mayoría eran conjuntos de datos utilizados en las evaluaciones SemEval. También hubo estudios que utilizaron tweets que expresaban opiniones, comentarios de redes sociales como Weibo y hubo uno que utilizó noticias financieras.

Y para vectorizar los conjuntos de datos mencionados se utilizaron principalmente los siguientes *embeddings* de palabras o frases:

- Wor2vec
- GloVe
- FastText
- Keras embedding
- Fine-tuning embeddings
- Tencent AI Lab Embedding (para chino)
- Positional embeddings
- BERT embeddings (pre-entrenado)

3.6.3.6 Targets.

¿En qué categorías se clasificaron los textos?

La tarea de clasificar los sentimientos dentro del análisis sentimental consiste en clasificar la polaridad de los textos según los sentimientos expresados en los mismos, y en los estudios analizados se evalúa si esta opinión fue positiva, negativa o si tiene una posición neutral. También hubo un estudio que se enfoca en la clasificación de textos incluyendo la clasificación de sentimientos evaluando niveles de satisfacción frente a algo con calificaciones del 1 al 5.

3.7 Conclusiones.

Los modelos basados en *transformers* cuentan con una arquitectura basada en el mecanismo de auto atención permitiéndoles centrarse en el uso de la atención para extraer las características de los textos y sus dependencias tanto de las entradas como de las salidas, para ello evita la recurrencia y aplica paralelización, haciéndolo un modelo muy eficiente. Si bien es un modelo que cuenta con muchas características buenas como las mencionadas

anteriormente puede ser mejorado dependiendo de las técnicas a las cuales se apunte, combinándolo con distintos tipos de *embeddings*, haciendo uso de modelos pre-entrenados (*transfer learning*), combinando su arquitectura con otras para explotar al máximo sus capacidades y así de diversas formas, por lo cual es un bastante versátil y al ser medianamente nuevo (desde el 2017) todavía se tiene buen camino para mejorar y explotar este modelo. En cuanto a su aplicación en el análisis sentimental es de mucha utilidad para este obteniendo muy buenos resultados en precisión comparado con modelos más clásicos de *deep learning* como son LSTM o GRU. Este puede ser aplicado a variaciones de análisis sentimental mucho más complejas como análisis sentimental de dominios cruzados o análisis sentimental basado en aspectos mejorado (AESA), los cuales realizan ya no solo la clasificación de las polaridades de los sentimientos de dominios en específico, sino pretenden generalizar un poco esto para poderlo aplicar a diversos dominios; además que permite que los textos no estén limitados por una sola etiqueta, sino que se analice todo el contexto textual facilitando la discusión de diversos tópicos y la expresión de más de una emoción. En ese sentido, hacer un modelo basado en *transformers* para el análisis sentimental de información pública extraída de noticias y redes sociales se muestra como una técnica muy adecuada y computacionalmente enriquecedora en el área de procesamiento de lenguaje natural.

Capítulo 4. Modelos de análisis sentimental para conjuntos de datos en español

4.1 Introducción.

El presente capítulo consistirá en la redacción de los resultados obtenidos al momento de desarrollar las tareas correspondientes para cumplir con el primer objetivo de la presente tesis, el cual es: “Implementar un modelo que haga uso de redes neuronales recurrentes para realizar un análisis de sentimiento sobre un conjunto de datos en español”. Esto consistirá en desarrollar los modelos de aprendizaje profundo que harán uso de redes neuronales *transformers* y que desarrollen la tarea de análisis sentimental en un conjunto de datos en español. Para la presente tesis se optó por hacer dos modelos debido a los dos tipos de conjuntos de datos que se abordarán, uno estará implementado y entrenado para trabajar con publicaciones de redes sociales, específicamente *tweets*, y el otro para artículos periodísticos, esto debido a las diferencias entre estos conjuntos de datos, como por ejemplo el lenguaje usado y la extensión de estos.

A continuación, se va a redactar paso a paso cómo se desarrollaron estos modelos, desde cómo empezaron a implementar, cómo se experimentó con estos y con el conjunto de datos para así ir obteniendo mejores resultados, hasta cómo se optimizaron para mejorar las métricas establecidas.

4.2 Resultados Alcanzados.

4.2.1 Arquitectura base de red neuronal.

El primer resultado del primer objetivo de la tesis consiste en la elección de una arquitectura base la cual será utilizada posteriormente para desarrollar los modelos de análisis sentimental que finalmente serán utilizados.

Se escogió como arquitectura base a BERT, la cual será entrenada luego mediante la técnica de *fine tuning* con conjuntos de datos que ya cuentan con una clasificación de la polaridad de los sentimientos expresados en dichos textos.

Para una mejor comprensión de la arquitectura seleccionada se explicará mediante un gráfico su funcionamiento y además se proporcionarán algunos conceptos clave para facilitar su entendimiento.

En primer lugar, se explicará en que consiste el *fine tuning*, dado que esto servirá para entender cómo se hará el entrenamiento de los modelos de la presente tesis. El *fine tuning* es una técnica muy usada en el *transfer learning*, que lo que hace es hacer uso de un modelo previamente entrenado para realizar alguna tarea en específico y aprovechar lo que este modelo aprendió previamente y aplicarlo a otra tarea similar.

En segundo lugar, se brindará información detallada acerca de BERT, la arquitectura base elegida. BERT es un modelo de representación de lenguaje cuyas siglas significan “Representación de codificador bidireccional de *transformers*”, y está diseñado para pre entrenar representaciones profundas y bidireccionales de texto sin clasificar mediante el reconocimiento conjunto del contexto de una palabra considerando tanto a la izquierda como la derecha de la oración en todas sus capas y finalmente como resultado de ello el modelo pre entrada de BERT puede ser ajustado, mediante *fine tuning*, con una sola capa adicional de salida y con ello crear modelos del estado del arte para una amplia gama de tareas, donde está incluida la clasificación de textos, por lo cual este puede ser aplicado para el análisis de sentimiento (huggingface, 2020).

A continuación, se mostrará una gráfica que ilustrará con un ejemplo cómo funciona el *embedding* de BERT y como se van vectorizando las palabras.

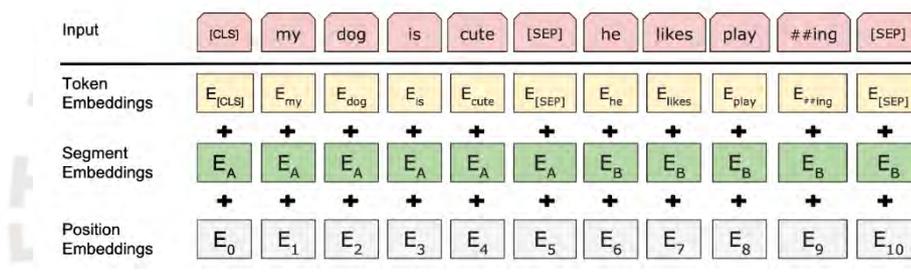
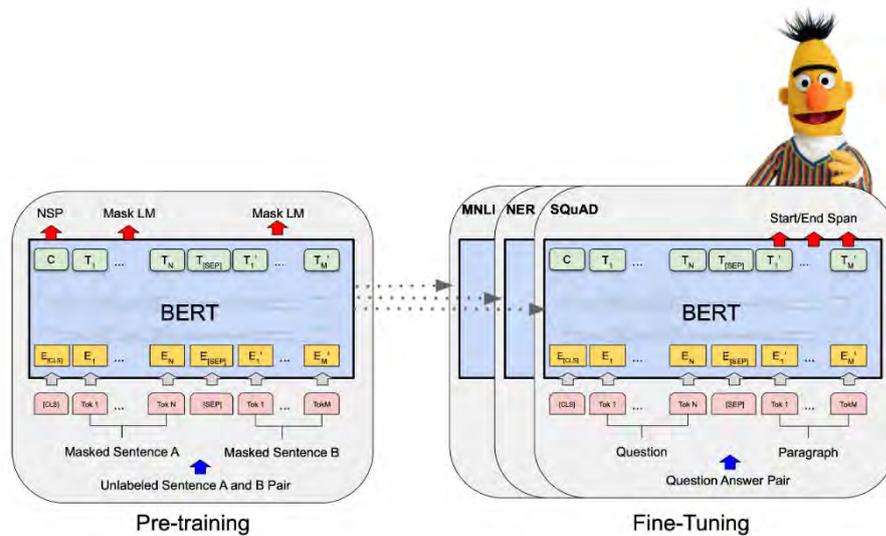


Figura 3. Diagrama de BERT. Tomado de COURSERA

En la imagen se puede ver que las palabras se irán codificando en tres instancias, primero por token, o palabra en sí, luego por oraciones, dado que BERT reconoce caracteres de separadores y finalmente se hace una codificación posicional según los contextos previamente extraídos. Además, se ve como mediante el *fine tuning* se realizan diversas tareas haciendo uso de BERT, como lo son inferencia del lenguaje natural (MNLI), reconocimiento de entidades (NER) y respuesta de preguntas.

Para la implementación de un primer modelo haciendo uso de la arquitectura base de red neuronal se decidió utilizar la librería *transformers*, la cual proporciona arquitecturas de uso general, como lo es BERT, para tareas de comprensión del lenguaje natural (NLU) y

generación del lenguaje (NLG), cuenta con una amplia cantidad de modelos previamente entrenados en más de 100 idiomas y cuentan con una interoperabilidad profunda entre TensorFlow 2.0 y PyTorch (huggingface, 2020).

Dentro de esta librería, se escogió como arquitectura de modelo base la denominada “*bert-base-uncased*”, el cual es un modelo de 12 capas, 768 ocultas, 12 cabeceras y cuenta con 110 millones de parámetros, entrenado en textos redactados en minúsculas en inglés (huggingface, 2020).

Para probar el modelo escogido se entrenó el mismo con un conjunto de datos denominado *SMILE Twitter Emotion dataset* (Wang *et al.*, 2016), que consiste en un conjunto de 3085 tweets en inglés categorizados según la emoción expresada.

El conjunto de datos fue preprocesado de tal forma que se eliminaron los tweets que expresaban múltiples sentimientos y los que estaban etiquetados como “*no code*”, dado que estos no aportarían al entrenamiento del modelo.

Posteriormente dicho conjunto de datos preprocesado fue particionado de la siguiente forma: 85% para el entrenamiento y 15% para la validación, de tal forma que se divida de esta forma dentro de cada categorización de sentimiento, ello para que no haya un desbalance en cuanto a cantidades por sentimiento en el conjunto de datos de entrenamiento o en el de validación, quedando la distribución que se mostrará en la tabla a continuación.

Tabla 11. Distribución del dataset SMILE luego del preprocesamiento y la partición

| Sentimiento | Cantidad Total | Cantidad para el entrenamiento | Cantidad para la validación |
|--------------|----------------|--------------------------------|-----------------------------|
| <i>happy</i> | 1137 | 966 | 171 |

| | | | |
|---------------------|------|------|-----|
| <i>non-relevant</i> | 214 | 182 | 32 |
| <i>angry</i> | 57 | 48 | 9 |
| <i>surprise</i> | 35 | 30 | 5 |
| <i>sad</i> | 32 | 27 | 5 |
| <i>disgust</i> | 6 | 5 | 1 |
| Total | 1481 | 1258 | 223 |

Luego de ello se procedió a tokenizar el conjunto de datos haciendo uso de BertTokenizer, perteneciente a la librería *transformers*, generando tensores de pyTorch, uno que contiene los inputs, otro las máscaras de atención y uno con las etiquetas de estos, y luego estos tres se usan para inicializar TensorDatasets, uno para el conjunto de datos de entrenamiento y otro para el conjunto de validación.

Después se procede a preparar el modelo pre entrenado de BERT. Como el análisis de sentimiento es una clasificación de textos según la polaridad que estos expresan se utilizará BertForSequenceClassification y se inicializa con el modelo previamente mencionado “*bert-base-uncased*”.

Una vez inicializado el modelo y teniendo los TensorDatasets se procede a crear los Data Loaders, los cuales servirán para entrenar el modelo.

También se definió AdamW como el optimizador a usar en el modelo y un planificador lineal para ajustar la tasa de aprendizaje del modelo a medida que pasa cada

época, cabe aclarar que posteriormente se realizará una experimentación con optimizadores y *schedulers* para así obtener la mejor combinación.

Para entrenar el modelo se decidió hacerlo durante 10 épocas e ir exportando cada modelo entrenado para luego probar cada uno y además se fueron guardando los valores de Validation Loss y de F1_Score por cada época.

Los resultados del entrenamiento fueron los que se mostrarán en la siguiente tabla.

Tabla 12. Resultados de entrenamiento del modelo base

| N° Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
|----------|-----------------|--------------|-------------------------------|
| 1 | 0,7342 | 0,8465 | 69 |
| 2 | 0,7266 | 0,8393 | 69 |
| 3 | 0,7738 | 0,8506 | 69 |
| 4 | 0,7663 | 0,8604 | 69 |
| 5 | 0,8291 | 0,8608 | 68 |
| 6 | 0,8282 | 0,8641 | 68 |
| 7 | 0,8189 | 0,8645 | 68 |
| 8 | 0,8239 | 0,8595 | 68 |
| 9 | 0,8198 | 0,8595 | 68 |
| 10 | 0,8217 | 0,8595 | 68 |
| | | Total | 684 |

Finalmente se definió una función que servirá para evaluar la precisión de los modelos entrenados con lo que se probarán los modelos generados.

Se eligió el generado en la época 7 para ser evaluado, por el hecho de que en esta se obtuvo el *F1 Score* más alto, con lo que se obtuvieron los resultados mostrados en la tabla a continuación.

Tabla 13. Resultados de evaluación del modelo base

| Sentimiento | Aciertos | Total | Porcentaje |
|--------------|------------|------------|---------------|
| happy | 162 | 171 | 94,74% |
| non-relevant | 20 | 32 | 62,50% |
| angry | 6 | 9 | 66,67% |
| disgust | 0 | 1 | 0,00% |
| sad | 3 | 5 | 60,00% |
| surprise | 2 | 5 | 40,00% |
| Total | 193 | 223 | 86,55% |

Finalmente con estos resultados del modelo evaluado, que fue el generado en la época 7 y tiene como nombre `BERT_ft_epoch7.model`, se puede ver que se tuvieron muy buenos resultados en la clasificación de textos que denotan alegría, resultados buenos en textos donde no se expresa un sentimiento relevante, en los que denotan enojo y tristeza, en los que denotan sorpresa se tuvo un resultado regular y finalmente en los que denotan disgusto se tuvieron muy malos resultados, pero en conjunto, el modelo tuvo un buen score de precisión, obteniendo un 86,55% de aciertos en las predicciones realizadas.

El modelo computacional desarrollado para poder obtener este resultado esperado se encuentra implementado en un archivo de Jupyter Notebook, el cual tiene como nombre `BaseModel.ipynb` y la ruta para acceder a este se encuentra mostrada en el Anexo C.

Y adicionalmente al modelo computación se desarrolló un reporte de la arquitectura base elegida donde se plasman los resultados obtenidos al entrenar y validar este modelo, y además se detalla la información del set de datos utilizado, el cual se encontrará mostrado en el Anexo D.

4.2.2 Arquitectura base entrenada y validada con un conjunto de datos en español.

El segundo resultado del primer objetivo de la presente tesis consiste en la implementación de un modelo haciendo uso de la arquitectura base previamente planteada, con la diferencia de que esta vez se trabaje con un conjunto de datos en español, por lo cual se vio por conveniente hacer unos cambios con respecto al modelo, pero manteniendo la arquitectura base. Se decidió hacer el cambio de BERT a BETO (“dccuchile/bert-base-spanish-wwm-cased”), el cual también es conocido como BERT en español, que consiste en un modelo entrenado sobre un gran corpus de datos en español. BETO es de tamaño similar a BERT-base y fue entrenado con la técnica *Whole Word Masking* (Departamento de Ciencias de la Computación Universidad de Chile, 2020). Con ello el cambio realizado es mínimo, teniendo como principal diferencia que BETO esta pre entrenado con un conjunto de datos en español, pero de ahí sigue la misma arquitectura que BERT, por ello este funcionaría mejor para el modelo que se desea implementar.

La implementación del modelo que hará uso de la arquitectura base con las modificaciones respectivas para que funcione correctamente con un conjunto de datos en español se realizó en un *notebook* de Jupyter que tiene como nombre SpanishModel.ipynb, donde se realizó toda la programación respectiva y será explicada a continuación.

Para entrenar dicho el segundo modelo base se utilizó el conjunto de datos usado en la competición del 2019 del Taller de Análisis Semántico en la SEPLN (TASS), el cual se

compone por 4 archivos de tweets en español correspondientes a países, Costa Rica, España, Perú y Uruguay, que se juntaron en un solo bloque de datos que resultó en un total de 3811 tweets, clasificados según la polaridad expresada, que podía ser positivo (P), negativo (N), neutral (NEU) y ninguno (NONE).

El conjunto de datos fue procesado de forma que se eliminaron los tweets que no tenían clasificación (NONE) y luego de ello se realizó una partición igual a la hecha en el modelo base presentado previamente, donde se separó el 85% del conjunto de datos para el entrenamiento y el 15% para la validación, quedando la distribución que se mostrará en la tabla a continuación.

Tabla 14. Distribución del dataset TASS 2019 luego del preprocesamiento y la partición

| Sentimiento | Cantidad Total | Cantidad para el entrenamiento | Cantidad para la validación |
|-------------|----------------|--------------------------------|-----------------------------|
| N | 1379 | 845 | 534 |
| NEU | 593 | 346 | 247 |
| P | 1081 | 638 | 443 |
| Total | 3053 | 1829 | 1224 |

Cabe destacar que los textos de los tweets no fueron preprocesados para esta instancia de prueba del modelo, dado que se desea ver cómo irá evolucionando este a medida que se realicen distintos procesamientos en los textos posteriormente.

Luego el conjunto de datos fue *tokenizado* haciendo uso de BertTokenizer, pero esta vez este fue cargado con BETO, y luego con ello se generaron los DataLoaders que servirán

como entrada para el modelo, siguiendo el mismo procedimiento realizado en el primer resultado del segundo objetivo de la tesis.

De igual forma que la arquitectura base se definió AdamW como el optimizador a usar en el modelo y un planificador lineal para ajustar la tasa de aprendizaje del modelo a medida que pasa cada época.

El entrenamiento se realizó por 20 épocas haciendo uso de la GPU como entorno de ejecución y se obtuvieron los resultados que serán mostrados a continuación.

Tabla 15. Resultados de entrenamiento del modelo base en español

| N° Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
|----------|-----------------|----------|-------------------------------|
| 1 | 0,8045 | 0,6084 | 23 |
| 2 | 0,7450 | 0,6314 | 23 |
| 3 | 0,7274 | 0,6906 | 23 |
| 4 | 0,7549 | 0,7050 | 23 |
| 5 | 0,8889 | 0,6960 | 23 |
| 6 | 0,9814 | 0,6834 | 23 |
| 7 | 1,0317 | 0,7095 | 23 |
| 8 | 1,1175 | 0,7003 | 23 |
| 9 | 1,3364 | 0,6884 | 23 |
| 10 | 1,4019 | 0,6838 | 23 |
| 11 | 1,3875 | 0,7014 | 23 |
| 12 | 1,4891 | 0,7037 | 23 |
| 13 | 1,5609 | 0,7026 | 23 |

| | | | |
|-----------|--------|--------------|-----|
| 14 | 1,6164 | 0,7041 | 23 |
| 15 | 1,6437 | 0,7055 | 23 |
| 16 | 1,7351 | 0,6996 | 23 |
| 17 | 1,8190 | 0,6837 | 23 |
| 18 | 1,7795 | 0,7018 | 23 |
| 19 | 1,8017 | 0,6975 | 23 |
| 20 | 1,8144 | 0,7029 | 23 |
| | | Total | 460 |

Finalmente haciendo uso de la misma función de evaluación se probaron los modelos generados, quedando como el mejor el generado en la época 7 dado que este obtuvo el *F1 Score* más alto de todas las épocas. A continuación, se mostrarán los resultados de manera detallada por sentimiento.

Tabla 16. Resultados de evaluación del modelo base

| Sentimiento | Aciertos | Total | Porcentaje |
|--------------------|-----------------|--------------|-------------------|
| P | 341 | 443 | 76,98% |
| N | 407 | 534 | 76,22% |
| NEU | 114 | 247 | 46,15% |
| Total | 862 | 1224 | 70,42% |

De estos resultados se puede ver que el modelo funcionó de muy buena manera clasificando textos que expresan sentimientos positivos y negativos, en cambio con textos neutrales se tuvo un acierto de menos del 50%, pero a nivel global se tuvo un score de acierto de 70.42%, lo cual para un primer modelo sin previo procesamiento del conjunto de datos es un resultado aceptable que se deberá ir mejorando.

El modelo computacional desarrollado para poder obtener este resultado esperado se encuentra implementado en un archivo de Jupyter Notebook, el cual tiene como nombre *Spanish_Model.ipynb* y la ruta para acceder a este se encuentra mostrada en el Anexo C.

Y adicionalmente al modelo computación se desarrolló un reporte donde se plasman los resultados obtenidos al entrenar y validar este modelo hecho para un conjunto de datos en español, y además se detalla la información del set de datos utilizado, el cual se encontrará mostrado en el Anexo E.

4.2.3 Modelo para análisis sentimental considerando arquitectura base.

El tercer resultado del primer objetivo de la tesis consiste en la implementación de un modelo para análisis sentimental considerando arquitectura base entrenada previamente elegida. Para la obtención del presente resultado se decidió desarrollar dos modelos distintos pero ambos basándose en la arquitectura base previamente elegida y explicada, la cual como se mencionó utilizará el modelo “*dccuchile/bert-base-spanish-wwm-uncased*”, también conocido como BETO, que cuenta con la misma arquitectura base del modelo utilizado para la obtención de la arquitectura del modelo base para un conjunto de datos en español, con la variación de que este no es *cased*, es decir, trabajará sin hacer distinción entre caracteres en mayúscula o en minúscula, los motivos de dicho cambio se presentarán posteriormente

cuando se detallan los experimentos realizados sobre los modelos, mostrando los resultados que se fueron obteniendo para así fundamentar los cambios realizados.

Se desarrollaron dos modelos, el primer modelo servirá para realizar la tarea de análisis sentimental a publicaciones de redes sociales, de manera específica para *tweets* en español, y el segundo modelo funcionará con artículos periodísticos. Se decidió trabajar de la siguiente manera por la diferencia de características existentes entre *tweets* y artículos periodísticos, como por ejemplo que el primero consiste en textos cortos que expresan opiniones, pero con lenguaje más coloquial, en cambio los artículos son textos más largos que no necesariamente van a expresar de manera tan clara una opinión y además están redactados con un lenguaje formal.

4.2.3.1 Modelo para análisis sentimental de tweets.

Para el desarrollo del modelo que realice análisis sentimental en tweets se trabajó sobre la arquitectura base para un conjunto de datos en español desarrollada para la obtención del segundo resultado del segundo objetivo, donde se realizaron una serie de pruebas sobre un mismo conjunto de datos, el cual es el mismo que se utilizó previamente, que es el conjunto de datos usados en la competición del 2019 del Taller de Análisis Semántico en la SEPLN (TASS), para con ello ir obteniendo mejores resultados de precisión.

A continuación, se mostrará una tabla que detalla los resultados obtenidos en base a las modificaciones realizadas en el preprocesamiento del conjunto de datos y también al cambio entre modelos previamente explicado, del *cased* por el *uncased*, el cual finalmente fue escogido como modelo final.

Tabla 17. Resultados de experimentación para la selección del modelo

| Nombre del Modelo | Nombre del archivo del código | Modelo pre entrenado usado | Preprocesamiento realizado sobre el conjunto de datos | Score de precisión máximo obtenido |
|---------------------|-------------------------------|-------------------------------|--|------------------------------------|
| Modelo Base Español | Spanish_Model.ipynb | bert-base-spanish-wwm-cased | No se realizó ningún preprocesamiento | 70,42% |
| Modelo Español 2 | Spanish_Model_2.ipynb | bert-base-spanish-wwm-cased | Se eliminaron los caracteres no alfanuméricos, se mantuvieron las tildes, se trabajó con mayúsculas y minúsculas | 72,93% |
| Modelo Español 3 | Spanish_Model_3.ipynb | bert-base-spanish-wwm-uncased | Se eliminaron los caracteres no alfanuméricos, se mantuvieron las tildes, se convirtió todo a minúsculas | 75,98% |
| Modelo Español 4 | Spanish_Model_4.ipynb | bert-base-spanish-wwm-uncased | Se eliminaron los caracteres no alfanuméricos, se eliminaron tags y hashtags, se mantuvieron las tildes, se convirtió todo a minúsculas | 73,14% |
| Modelo Español 5 | Spanish_Model_5.ipynb | bert-base-spanish-wwm-uncased | Se eliminaron los caracteres no alfanuméricos, se eliminaron sólo los tags de usuarios, se mantuvieron las tildes y se convirtió todo a minúsculas | 73,58% |

También se mostrará una gráfica que muestra a detalle como varía el F1 Score que se va obteniendo en cada época al momento de ir entrenando los modelos para así ver cómo se comportaron estos.

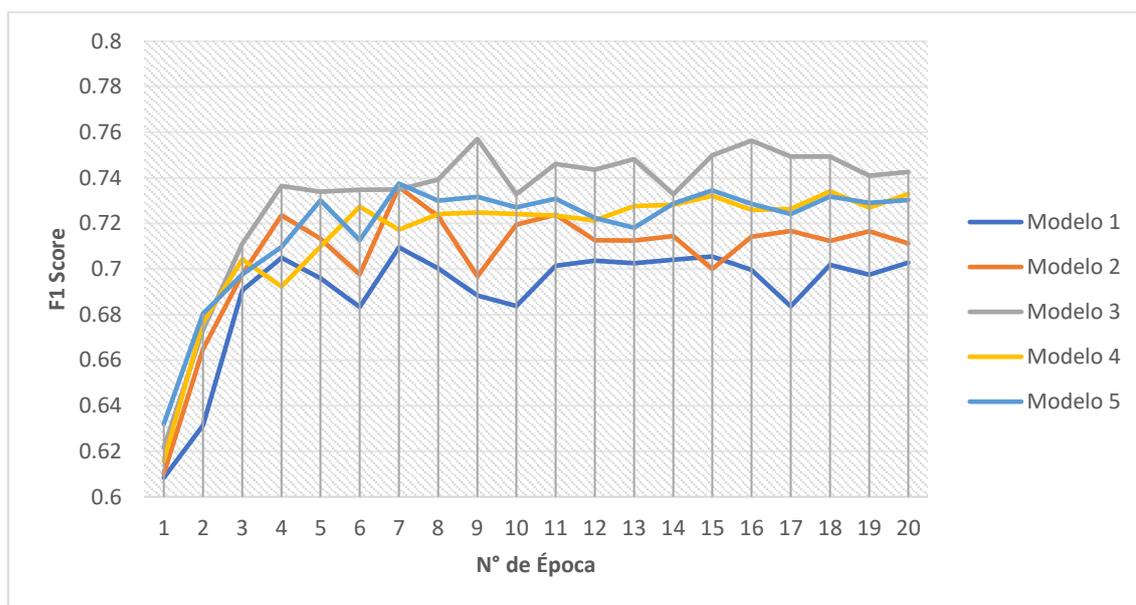


Figura 4. Gráfica de resultados de experimentación en el preprocesamiento para los modelos

De los resultados obtenidos en la experimentación realizada se puede ver que se obtuvo el mejor resultado de precisión en el tercer modelo realizado, trabajando con “*bert-base-spanish-wwm-uncased*” y eliminando los caracteres no alfanuméricos. Este score se obtuvo en el modelo generado en la novena época de entrenamiento, y en la evaluación respectiva de dicho modelo se obtuvieron los resultados que se mostrarán en la tabla a continuación.

Tabla 18. Resultados de evaluación del Modelo Español para tweets

| Sentimiento | Aciertos | Total | Porcentaje |
|-------------|----------|-------|------------|
| P | 124 | 162 | 76,54% |
| N | 181 | 207 | 87,44% |

| | | | |
|--------------|------------|------------|---------------|
| NEU | 43 | 89 | 48,31% |
| Total | 348 | 458 | 75,98% |

El modelo computacional desarrollado para poder obtener este resultado esperado se encuentra implementado en un archivo de Jupyter Notebook, el cual tiene como nombre *Spanish_Model_3.ipynb* y la ruta para acceder a este se encuentra mostrada en el Anexo C, además en dicho repositorio se encuentran los demás modelos utilizados para experimentación y elección del modelo final.

Y adicionalmente al modelo computación se desarrolló un reporte donde se plasman los resultados obtenidos al entrenar y validar este modelo hecho para tweets en español, y además se detalla la información del set de datos utilizado, el cual se encontrará mostrado en el Anexo F.

4.2.3.2 Modelo para análisis sentimental de artículos periodísticos.

Para el desarrollo del modelo computacional que realice el análisis sentimental para artículos periodísticos se utilizaron los resultados de la experimentación realizada para en el modelo desarrollado para tweets. Es por ello por lo que este se desarrolló haciendo uso del modelo pre entrenado de “bert-base-spanish-wwm-uncased” y se entrenó con un conjunto de datos etiquetado proporcionado por una empresa dedicada al monitoreo de medios. Este conjunto de datos con el cual se entrenará el presente modelo consiste en textos que a medida de resumen donde se expresan sentimientos hacia entidades en concreto, se usaron 4 bases de datos, una donde se hablaba del Ministerio de Cultura, la Clínica Internacional, la Defensoría del Pueblo y de RIMAC Seguros, es por ello por lo que el presente modelo

funcionará correctamente con textos similares a estos. A continuación, se mostrará una imagen de la estructuración de dicho conjunto de datos.

| | Título | Resumen | Tono |
|----------|--|---|-------------|
| 0 | Machu Picchu mantiene sus tarifas para este 2020 | A solicitud de la Dirección Desconcentrada de ... | Positivo |
| 1 | Melina León: "Soy de la generación que sacó de..." | Melina León estrenará este año Canción sin nom... | Positivo |
| 2 | Producción peruana Alzira será premiada en España | Alzira de Giuseppe Verdi, la primera coproduc... | Positivo |
| 3 | Avance del 2019: Minedu estableció alfabetos d... | El Ministerio de Educación (Minedu) logró en e... | Positivo |
| 4 | Caso Sheraton: todos los detalles sobre proces... | El hotel Sheraton ubicado en la cuadra 6 de la... | Neutro |

Figura 5. Conjunto de datos de artículos periodísticos

Luego de ello se procedió a limpiar el conjunto de datos eliminando columnas vacías, y realizando un balanceo en el conjunto, quedando finalmente un total de 3384 artículos resumidos con la siguiente distribución de tonos o sentimientos, expresados en el mismo: 1128 positivos, 1128 neutros y 1128 negativos.

Con dicho conjunto de datos se procedió a dividirlo en un conjunto para el entrenamiento y otro para la validación, quedando la distribución que se mostrará en la tabla a continuación.

Tabla 19. Distribución del dataset de artículos periodísticos luego del preprocesamiento y la partición

| Sentimiento | Cantidad Total | Cantidad para el entrenamiento | Cantidad para la validación |
|--------------------|-----------------------|---------------------------------------|------------------------------------|
| Negativo | 1128 | 959 | 169 |
| Neutro | 1128 | 959 | 169 |
| Positivo | 1128 | 959 | 169 |
| Total | 3384 | 2877 | 507 |

Con ello se procedió a armar los *tokenizar* el conjunto de datos y con ello armar los *DataLoaders*, para entrenar el modelo establecido. Este modelo al igual que los base utilizó como optimizador AdamW con una tasa de aprendizaje de $1e-5$, y se definió un planificador lineal que va alterando dicha tasa a medida que pasan las épocas.

Finalmente se entrenó el modelo por 20 épocas obteniendo los resultados mostrados a continuación.

Tabla 20. Resultados de entrenamiento del modelo en español para artículos periodísticos

| N° Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
|----------|-----------------|----------|-------------------------------|
| 1 | 0,7446 | 0,6765 | 220 |
| 2 | 0,6639 | 0,7177 | 221 |
| 3 | 0,6428 | 0,7512 | 221 |
| 4 | 0,7373 | 0,7269 | 222 |
| 5 | 0,8152 | 0,7425 | 221 |
| 6 | 0,9083 | 0,7449 | 221 |
| 7 | 1,0300 | 0,7441 | 221 |
| 8 | 1,1497 | 0,7375 | 221 |
| 9 | 1,2353 | 0,7347 | 221 |
| 10 | 1,4261 | 0,7204 | 221 |
| 11 | 1,4016 | 0,7401 | 221 |
| 12 | 1,3465 | 0,7527 | 221 |
| 13 | 1,4981 | 0,7399 | 221 |
| 14 | 1,481 | 0,7528 | 221 |

| | | | |
|----|--------|--------------|-------|
| 15 | 1,5984 | 0,7343 | 221 |
| 16 | 1,5618 | 0,75 | 221 |
| 17 | 1,5309 | 0,7585 | 221 |
| 18 | 1,5827 | 0,7553 | 221 |
| 19 | 1,5947 | 0,7575 | 221 |
| 20 | 1,6146 | 0,7461 | 221 |
| | | Total | 4420* |

*El resultado total del tiempo de entrenamiento es de 1 hora 13 minutos y 40 segundos

De los resultados obtenidos se ve que el mejor modelo se obtuvo en la época 17 de entrenamiento, donde se obtuvo el F1 Score más alto, por ello se decidió evaluar dicho modelo generado obteniendo los resultados de evaluación mostrados en la tabla a continuación.

Tabla 21. Resultados de evaluación del modelo en español para artículos periodísticos

| Sentimiento | Aciertos | Total | Porcentaje |
|--------------|------------|------------|---------------|
| Positivo | 147 | 169 | 86,98% |
| Neutro | 125 | 170 | 73,53% |
| Negativo | 161 | 169 | 95,27% |
| Total | 433 | 508 | 85,24% |

Con los resultados de la evaluación, se puede ver que con dicho modelo desarrollado se obtuvo un score de precisión de 85,24% en el modelo.

El modelo computacional desarrollado para poder obtener este resultado esperado se encuentra implementado en un archivo de Jupyter Notebook, el cual tiene como nombre *Spanish_Model_News.ipynb* y la ruta para acceder a este se encuentra mostrada en el Anexo C.

Y adicionalmente al modelo computación se desarrolló un reporte donde se plasman los resultados obtenidos al entrenar y validar este modelo hecho para artículos en español, y además se detalla la información del set de datos utilizado, el cual se encontrará mostrado en el Anexo G.

4.2.4 Modelo para análisis sentimental optimizado.

Para el cumplimiento del cuarto resultado del primer objetivo de la presente tesis se implementaron dos modelos debidamente optimizados que realicen la tarea de análisis de sentimiento para conjuntos de datos en español, uno para tweets y otro para artículos periodísticos.

La optimización se desarrolló sobre los modelos implementados para el tercer resultado del segundo objetivo y para poder escoger la mejor forma de optimizar estos modelos se realizó una experimentación dichos modelos. Esta consistía en probar en primera instancia dos distintos tipos de optimizadores para redes neuronales *transformers*, implementados para Python por la librería de *huggingface*, los cuales son *Adafactor* y *AdamW*, y combinando estos optimizadores con cinco tipos de *schedulers* (planificadores), para ajustar la tasa de aprendizaje del modelo a medida que pasa cada época, y finalmente comparar el comportamiento de los modelos a medida que estos se van entrenando con cada

una de estas combinaciones para así escoger la mejor forma de optimizar los modelos. Cabe destacar que los modelos desarrollados para los resultados 1,2 y 3 del presente objetivo utilizaban como optimizador *AdamW* y como *scheduler* el *linear_schedule_with_warmup*.

Para la experimentación se desarrollaron un total de 20 modelos, 10 para tweets y 10 para artículos periodísticos, donde 10 utilizaban Adafactor y 10 AdamW, combinándolos con los siguientes schedulers:

- Ajuste constante (*constant_schedule*)
- Ajuste constante con calentamiento (*constant_schedule_with_warmup*)
- Ajuste de coseno con calentamiento (*cosine_schedule_with_warmup*)
- Ajuste coseno con reinicios forzados y calentamiento (*cosine_with_hard_restarts_schedule_with_warmup*)
- Ajuste lineal con calentamiento (*linear_schedule_with_warmup*)

4.2.4.1 Experimentación sobre modelo de tweets.

Como se mencionó anteriormente la experimentación se realizó sobre los modelos del tercer resultado, por ello el conjunto de datos usados siguió siendo el *dataset* de TASS 2019, y este se preprocesó siguiendo el mejor resultado obtenido en la experimentación para la obtención del segundo resultado, el cual tenía como preprocesamiento el eliminar los caracteres no alfanuméricos del texto, mantener las tildes y convertir todo el texto a minúsculas, se mantuvo a BETO uncased (*bert-base-spanish-wwm-uncased*) como el modelo pre entrenado a usar y se siguió la misma metodología de implementación usada en resultados anteriores.

Los resultados obtenidos en la experimentación realizada se encuentran mostrados en la tabla a continuación.

Tabla 22. Resultados de experimentación para la optimización del modelo para publicaciones de redes sociales

| Nombre de Modelo | Optimizador | Scheduler | Mejor score de precisión obtenido |
|------------------|-------------|--|-----------------------------------|
| Modelo 1 | Adafactor | constant_schedule | 45,41% |
| Modelo 2 | | constant_schedule_with_warmup | 45,85% |
| Modelo 3 | | cosine_schedule_with_warmup | 43,45% |
| Modelo 4 | | cosine_with_hard_restarts_schedule_with_warmup | 33,41% |
| Modelo 5 | | linear_schedule_with_warmup | 32,75% |
| Modelo 6 | AdamW | constant_schedule | 74,89% |
| Modelo 7 | | constant_schedule_with_warmup | 73,36% |
| Modelo 8 | | cosine_schedule_with_warmup | 76,42% |
| Modelo 9 | | cosine_with_hard_restarts_schedule_with_warmup | 71,40% |
| Modelo 10 | | linear_schedule_with_warmup | 75,98% |

En la tabla se muestra el mejor score de precisión obtenido en cada variación realizada sobre los modelos, de donde se puede ver que al cambiar el optimizador a *Adafactor* se reduce considerablemente el score de precisión casi a la mitad de lo obtenido al momento de usar *AdamW*, por lo que *Adafactor* se descarta. Y en cuanto a los distintos *schedulers* si bien las variaciones no son muy altas, se ve que el mejor score de precisión se obtiene haciendo uso

del ajuste de coseno con calentamiento (*cosine_schedule_with_warmup*), el cual obtiene un score de precisión de 76,42%, mejorando en un 0,44% el utilizado en los resultados previos (*linear_schedule_with_warmup*).

Además de la tabla previa, se realizó un gráfico que muestra la evolución del F1 Score obtenido en cada época de entrenamiento y por cada modelo desarrollado, el cual se muestra a continuación.

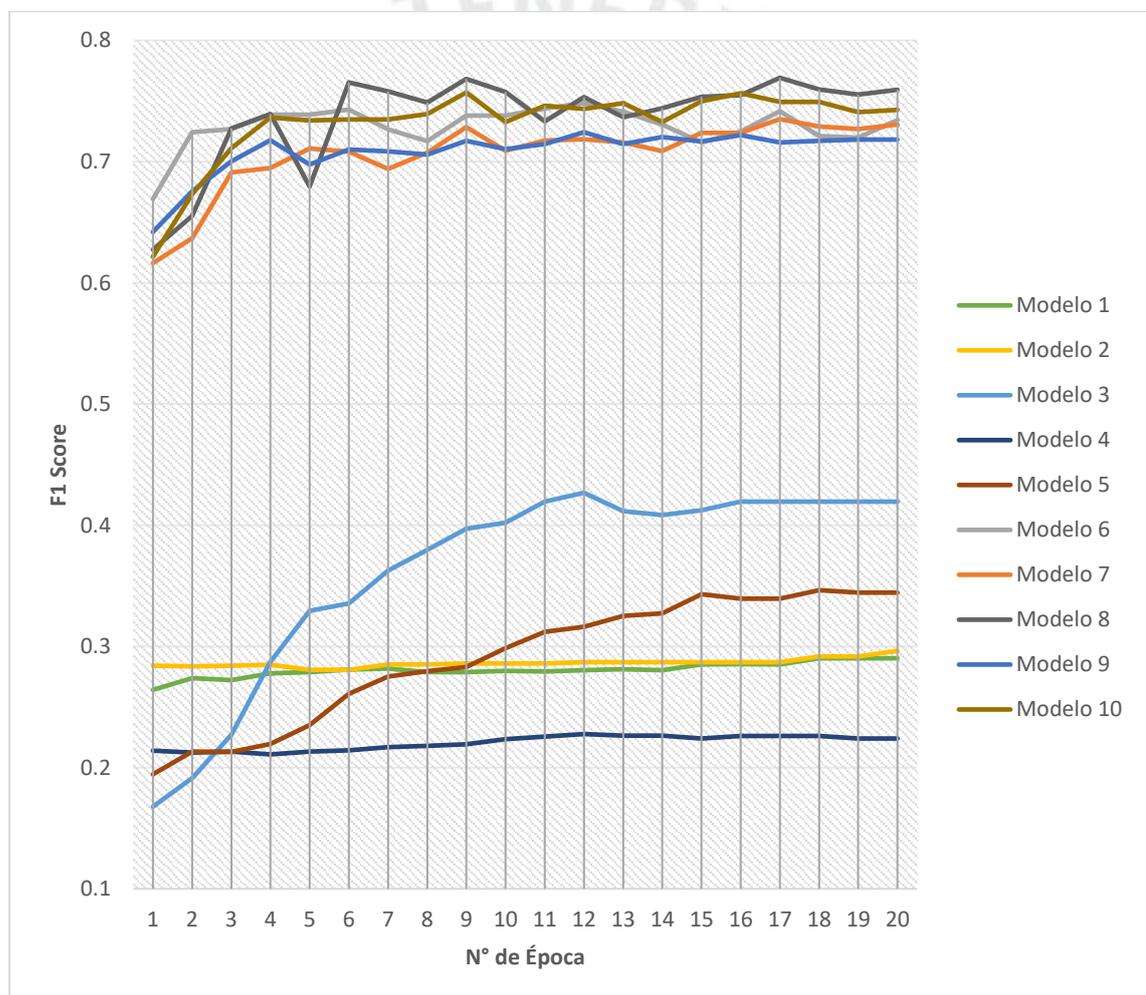


Figura 6. Gráfica de resultados de experimentación en los optimizadores de los modelos de publicaciones de redes sociales

En el gráfico se resalta claramente lo observado en la tabla, que es que, con los 5 primeros modelos desarrollados, los cuales utilizan Adafactor como optimizador el desempeño del modelo es menor que al hacer uso de AdamW, el cual usan los modelos del 6 al 10. Y además se ve que el pico más alto de F1 Score lo obtiene el modelo 8 en la época 17 de entrenamiento, por lo cual se confirma que la mejor combinación de optimizadores y schedules para el modelo desarrollado será AdamW con ajuste de coseno con calentamiento (*cosine_schedule_with_warmup*).

4.2.4.2 Experimentación sobre modelo de artículos periodísticos

La experimentación para la optimización del modelo de artículos periodísticos se realizó, al igual que en el de tweets, sobre el modelo del segundo resultado, el cual tenía como preprocesamiento el eliminar los caracteres no alfanuméricos del texto, mantener las tildes y convertir todo el texto a minúsculas, se mantuvo a BETO uncased (bert-base-spanish-wwm-uncased) como el modelo pre entrenado a usar y se siguió la misma metodología de implementación usada en resultados anteriores. Además, cabe recalcar que se realizó sobre el conjunto de datos brindado por la empresa de monitoreo de medios.

Los resultados obtenidos en la experimentación realizada se encuentran mostrados en la tabla a continuación.

Tabla 23. Resultados de experimentación para la optimización del modelo para artículos periodísticos

| Nombre de Modelo | Optimizador | Scheduler | Mejor score de precisión obtenido |
|------------------|-------------|-------------------------------|-----------------------------------|
| Modelo 1 | Adafactor | constant_schedule | 78,94% |
| Modelo 2 | | constant_schedule_with_warmup | 83,07% |
| Modelo 3 | | cosine_schedule_with_warmup | 85,04% |

| | | | |
|-----------|-------|--|--------|
| Modelo 4 | | cosine_with_hard_restarts_schedule_wit h_warmup | 88,39% |
| Modelo 5 | | linear_schedule_with_warmup | 81,10% |
| Modelo 6 | AdamW | constant_schedule | 79,33% |
| Modelo 7 | | constant_schedule_with_warmup | 80,91% |
| Modelo 8 | | cosine_schedule_with_warmup | 79,72% |
| Modelo 9 | | cosine_with_hard_restarts_schedule_wit h_warmup | 77,76% |
| Modelo 10 | | linear_schedule_with_warmup | 85,24% |

En la tabla se muestra el mejor score de precisión obtenido en cada variación realizada sobre los modelos, de donde se puede ver que los valores no varían tanto entre los dos optimizadores, por lo cual la elección del mejor modelo dependerá de la combinación con el *scheduler*.

Se observa que el mejor score de precisión se obtiene haciendo uso de Adafactor con ajuste de coseno con reinicios forzados y calentamiento (*cosine with hard restarts schedule with warmup*), el cual obtiene un score de precisión de 88,39%, mejorando en un 3,15% el utilizado en los resultados previos (AdamW con *linear_schedule_with_warmup*).

También se realizó un gráfico que muestra la evolución del F1 Score obtenido en cada época de entrenamiento y por cada modelo desarrollado, el cual se muestra a continuación.

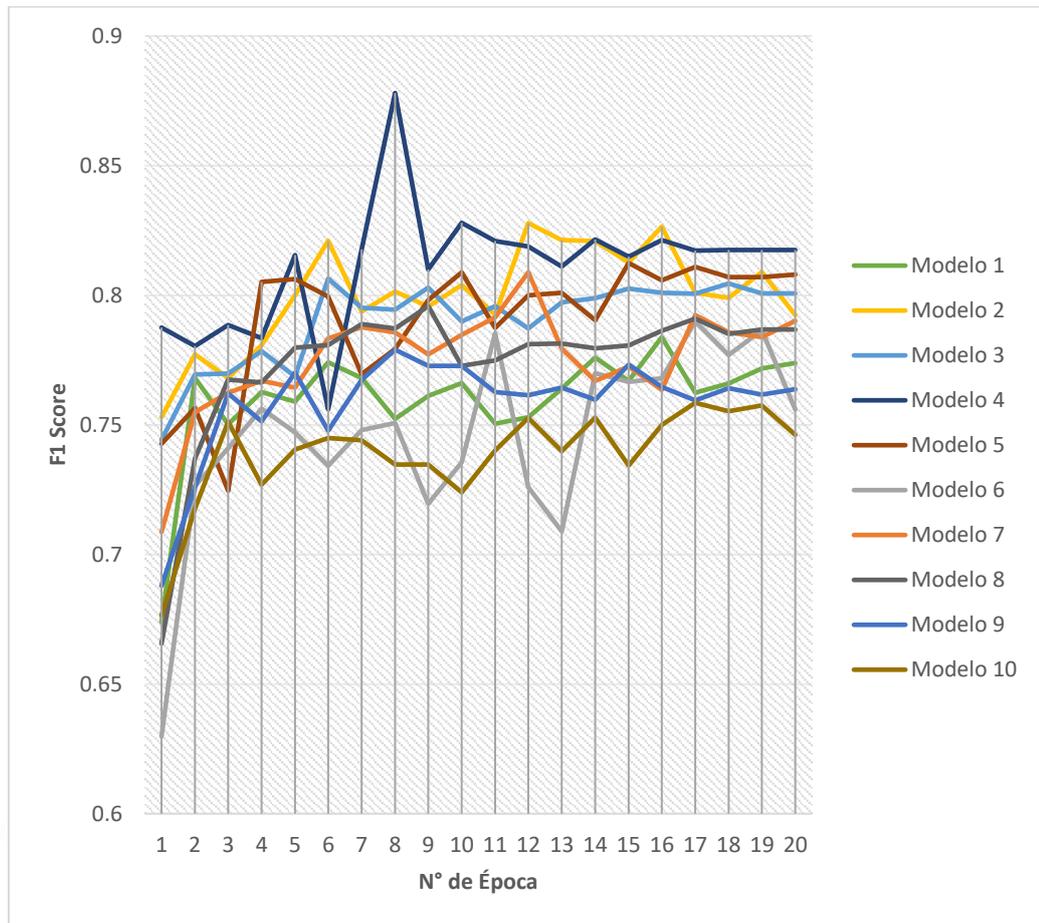


Figura 7. Gráfica de resultados de experimentación en los optimizadores de los modelos para artículos periodísticos

En el gráfico se recalca la poca diferencia del valor del F1 Score entre los modelos con los distintos optimizadores y *schedulers*.

Se resalta que el pico más alto de F1 Score lo obtiene el modelo 4 en la época 8 de entrenamiento, por lo cual se confirma que la mejor combinación de optimizadores y *schedulers* para el modelo desarrollado será Adafactor con ajuste de coseno con reinicios forzados y calentamiento (*cosine with hard restarts schedule with warmup*).

4.2.4.3 Modelo optimizado para análisis sentimental de tweets.

El modelo optimizado para tweets fue el octavo modelo desarrollado en la experimentación de optimizadores de tweets, el cual fue implementado de la misma forma que el modelo del tercer resultado con la diferencia del que como scheduler se utilizó el ajuste de coseno con calentamiento (*cosine_schedule_with_warmup*), dado que este ya venía utilizando AdamW como optimizador. Los resultados detallados del entrenamiento se muestran en la tabla a continuación.

Tabla 24. Resultados de entrenamiento del modelo en español optimizado para tweets

| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
|----------|-----------------|----------|-------------------------------|
| 1 | 0,7394 | 0,6274 | 14 |
| 2 | 0,6428 | 0,6552 | 14 |
| 3 | 0,6647 | 0,7274 | 14 |
| 4 | 0,6736 | 0,7394 | 14 |
| 5 | 0,9123 | 0,6796 | 14 |
| 6 | 0,8004 | 0,7655 | 14 |
| 7 | 0,9082 | 0,7580 | 14 |
| 8 | 1,0194 | 0,7488 | 14 |
| 9 | 1,0478 | 0,7684 | 14 |
| 10 | 1,1424 | 0,7575 | 14 |
| 11 | 1,2691 | 0,7332 | 14 |

| | | | |
|----|--------|--------------|-----|
| 12 | 1,2371 | 0,7532 | 14 |
| 13 | 1,2872 | 0,7366 | 14 |
| 14 | 1,3581 | 0,7441 | 14 |
| 15 | 1,3012 | 0,7536 | 14 |
| 16 | 1,4072 | 0,7549 | 14 |
| 17 | 1,3074 | 0,7692 | 14 |
| 18 | 1,3407 | 0,7595 | 14 |
| 19 | 1,3856 | 0,7553 | 14 |
| 20 | 1,3377 | 0,7592 | 14 |
| | | Total | 280 |

De los resultados obtenidos se ve que el mejor modelo se obtuvo en la época 17 de entrenamiento, donde se obtuvo el F1 Score más alto, por ello se decidió evaluar dicho modelo generado obteniendo los resultados de evaluación mostrados en la tabla a continuación.

Tabla 25. Resultados de evaluación del modelo en español optimizado para tweets

| Sentimiento | Aciertos | Total | Porcentaje |
|-------------|----------|-------|------------|
| P | 123 | 162 | 75,93% |
| N | 173 | 207 | 83,57% |

| | | | |
|--------------|------------|------------|---------------|
| NEU | 54 | 89 | 60,67% |
| Total | 350 | 458 | 76,42% |

Con los resultados de la evaluación, se puede ver que con dicho modelo desarrollado se obtuvo un score de precisión de 76.42% en el modelo, cumpliendo el IOV planteado para este resultado, dado que se planteó que se debía obtener un score de precisión mayor a 75%.

El modelo computacional desarrollado para poder obtener este resultado esperado se encuentra implementado en un archivo de Jupyter Notebook, el cual tiene como nombre *Spanish_Model_Tweets_Optimized.ipynb* y la ruta para acceder a este se encuentra mostrada en el Anexo C.

Y adicionalmente al modelo computación se desarrolló un reporte donde se plasman los resultados obtenidos al entrenar y validar este modelo optimizado hecho para artículos en español, y además se detalla la información del set de datos utilizado, el cual se encontrará mostrado en el Anexo H.

4.2.4.2 Modelo optimizado para análisis sentimental de artículos periodísticos.

Finalmente, el modelo optimizado implementado para artículos periodísticos siguió los resultados de la experimentación de optimizadores realizada para artículos periodísticos, por lo cual el modelo realizado para el resultado 3 se modificó, cambiando el optimizador y el *scheduler* a Adafactor con ajuste de coseno con reinicios forzados y calentamiento (*cosine with hard restarts schedule with warmup*), luego de eso se siguió la misma metodología usada y explicada en los resultados previos.

Los resultados detallados del entrenamiento se muestran en la tabla a continuación.

Tabla 26. Resultados de entrenamiento del modelo en español optimizado para artículos periodísticos

| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
|----------|-----------------|----------|-------------------------------|
| 1 | 0,5632 | 0,7875 | 230 |
| 2 | 0,5615 | 0,7804 | 230 |
| 3 | 0,6399 | 0,7885 | 230 |
| 4 | 0,7598 | 0,7834 | 230 |
| 5 | 0,8556 | 0,8155 | 230 |
| 6 | 1,3877 | 0,7560 | 230 |
| 7 | 1,0742 | 0,8177 | 230 |
| 8 | 1,2907 | 0,8780 | 229 |
| 9 | 1,1965 | 0,8099 | 229 |
| 10 | 1,2899 | 0,8279 | 229 |
| 11 | 1,1851 | 0,8209 | 229 |
| 12 | 1,2150 | 0,8188 | 229 |
| 13 | 1,3017 | 0,8110 | 229 |
| 14 | 1,2709 | 0,8215 | 229 |
| 15 | 1,3107 | 0,8148 | 229 |
| 16 | 1,3056 | 0,8213 | 229 |

| | | | |
|----|--------|--------------|-------|
| 17 | 1,3135 | 0,8172 | 229 |
| 18 | 1,3145 | 0,8174 | 229 |
| 19 | 1,3082 | 0,8174 | 227 |
| 20 | 1,3092 | 0,8174 | 226 |
| | | Total | 4582* |

*El resultado total del tiempo de entrenamiento es de 1 hora 16 minutos y 22 segundos

De los resultados obtenidos se ve que el mejor modelo se obtuvo en la época 8 de entrenamiento, donde se obtuvo el F1 Score más alto, por ello se decidió evaluar dicho modelo generado obteniendo los resultados de evaluación mostrados en la tabla a continuación.

Tabla 27. Resultados de evaluación del del modelo en español optimizado para artículos periodísticos

| Sentimiento | Aciertos | Total | Porcentaje |
|--------------|------------|------------|---------------|
| Positivo | 146 | 169 | 86,39% |
| Neutro | 141 | 170 | 82,94% |
| Negativo | 162 | 169 | 83,43% |
| Total | 449 | 508 | 88,39% |

Con los resultados de la evaluación, se puede ver que con dicho modelo desarrollado se obtuvo un score de precisión de 88,39% en el modelo.

El modelo computacional desarrollado para poder obtener este resultado esperado se encuentra implementado en un archivo de Jupyter Notebook, el cual tiene como nombre *Spanish_Model_News_Optimized.ipynb* y la ruta para acceder a este se encuentra mostrada en el Anexo C.

Y adicionalmente al modelo computación se desarrolló un reporte donde se plasman los resultados obtenidos al entrenar y validar este modelo hecho para artículos en español, y además se detalla la información del set de datos utilizado, el cual se encontrará mostrado en el Anexo I.

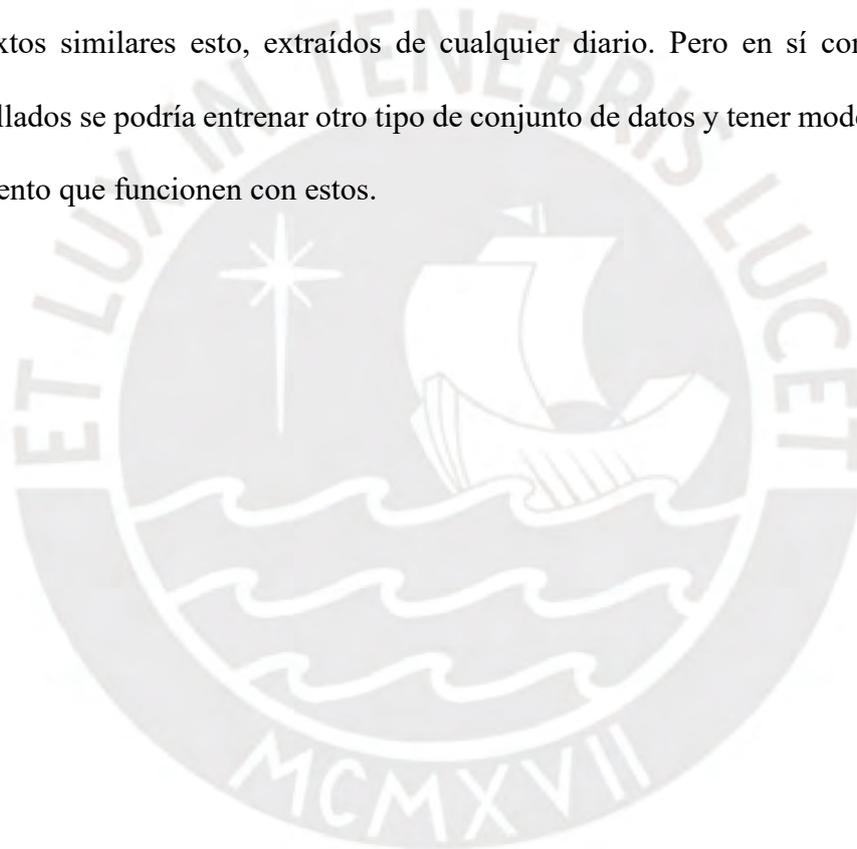
4.3 Discusión.

Para la implementación de un modelo que desarrolle análisis sentimental para un conjunto de datos en español, se escogió como arquitectura base el modelo preentrenado “*bert-base-uncased*”, el cual está desarrollado para funcionar con un conjunto de datos en inglés y exclusivamente puesta en minúsculas, el cual luego de ser entrenado con un set de datos de tweets en inglés retornó un score de precisión de 86.55%, por lo que se determinó que esta arquitectura si podía funcionar como modelo base. Por ello, para el segundo resultado establecido, el cual ya estipulaba que este sea entrenado con un conjunto de datos en español, se decidió usar una variación de este modelo el cual es denominado “*bert-base-spanish-wwm-cased*” o BETO (*Spanish BERT*), el cual contaba con las mismas características que la arquitectura base, pero este con la modificación de que fue entrenado con un conjunto de datos en español. Este segundo modelo fue entrenado y probado con el set de datos del TASS del 2019, el cual es un conjunto de tweets en español de España, Costa Rica, Perú y Uruguay, con el que sin ser preprocesado se obtuvo un score de precisión de 68.55%.

Luego para los casos de estudio planteados se decidió trabajar con dos modelos basados en la misma arquitectura, pero ambos entrenados con datasets de distintas características, el primero era para tweets en español, el cuál luego de realizar una serie de experimentos y pruebas con variaciones del modelo y con diversas formas de pre procesamiento, donde se obtuvo como resultado un modelo que hacía uso de “*bert-base-spanish-wwm-uncased*”, también conocido como BETO uncased, variación del modelo previamente usado con la diferencia que este no hace distinción entre mayúsculas y minúsculas, y se pre proceso la el conjunto de datos de tweets eliminando caracteres con alfanuméricos, con lo cual se mejoró el modelo, obteniendo un score de precisión 75,98%. Y en cuanto al segundo modelo, el cual funciona para artículos periodísticos, se trabajó con el mismo modelo y preprocesamiento de los textos, variando solo el conjunto de datos con el que se entrenó, obteniendo en este un score de precisión muy bueno de 89,33%.

Finalmente para la optimización del modelo se decidió probar entre dos optimizadores usados para redes neuronales Transformers, que son Adafactor y AdamW y 5 distintos tipos de schedulers para ajustes de la tasa de aprendizaje del modelo, para lo cual se experimentó con el modelo probando las distintas combinaciones de estos, con lo que se obtuvo que la mejor combinación a usar para tweets debía ser AdamW con ajuste de coseno con calentamiento (*cosine_schedule_with_warmup*) y para artículos periodísticos a Adafactor con ajuste de coseno con reinicios forzados y calentamiento (*cosine with hard restarts schedule with warmup*), dado que con ello el desempeño de los modelos implementados mejoraba con dichas combinaciones. Obteniendo así dos modelos optimizados para la realización del análisis de sentimiento, el modelo que trabaja con tweets tiene un score de precisión final de 76,42% y el modelo que trabaja sobre artículos periodísticos tiene un score de precisión final de 88,39%.

Los dos modelos que se obtienen finalmente como resultados funcionarán correctamente para textos pertenecientes al tipo de texto con el cual cada modelo fue entrenado, por ejemplo, para el modelo entrenado con tweets se podrá hacer uso de publicaciones de diversas redes sociales, como pueden ser Facebook, Twitter, Reddit, dado que estos tienen características similares, y para el caso del modelo de artículos periodísticos se tiene textos donde se expresan sentimientos acerca de alguna entidad en específico, por lo cual este funcionará para textos similares esto, extraídos de cualquier diario. Pero en sí con los modelos ya desarrollados se podría entrenar otro tipo de conjunto de datos y tener modelos de análisis de sentimiento que funcionen con estos.



Capítulo 5. Automatización de la definición de sentimientos de información pública de medios periodísticos y redes sociales

5.1 Introducción.

En el presente capítulo de la redacción de resultados se describirán detalladamente los pasos realizados para cumplir el tercer objetivo de la presente tesis, el cual consiste en la automatización de la definición de sentimientos expresados en información pública de medios periodísticos y de redes sociales, estandarizando así los criterios usados en la realización de dicha tarea. Para poder cumplir lo plasmado en dicho objetivo se realizará una aplicación web, la cual contará con una interfaz donde se podrán subir archivos a los cuales se les realizará el análisis de sentimiento, y esta enviará dichos archivos al *backend* de la aplicación, donde haciendo uso de los modelos previamente desarrollados en el objetivo dos, se realizarán dichas predicciones, y además se contará con una funcionalidad que demuestre de manera visual el funcionamiento de los modelos en la misma web, la cual le permitirá al usuario ingresar un texto al cual se le realizará el análisis sentimental. Y esta interfaz web permitirá como caso de estudio de la presente tesis el realizar un análisis de sentimiento a las bases de datos textuales que serán extraídas de internet en el último objetivo planteado, para con ello demostrar el funcionamiento de esta interfaz y del flujo a seguir.

5.2 Interfaz web para pruebas del usuario.

El resultado a obtener consiste en el desarrollo de una pequeña aplicación, la cual servirá para automatizar de la definición de sentimientos de información pública de medios periodísticos y redes sociales, para lo cual la principal funcionalidad con la que esta contará es la subida de archivos textuales con textos de medios periodísticos o publicaciones de redes sociales mediante la interfaz web de la misma, la cual enviará estos archivos al *backend* de

la aplicación, donde haciendo uso de los modelos de aprendizaje profundo previamente desarrollados se realizarán las inferencias de cada texto y se realizará la clasificación de la polaridad de los sentimientos expresada, generando con ello un archivo con la respuesta de esta clasificación.

Para la realización de la aplicación se desarrolló en primera instancia el *backend* de la misma. Este sería super simple dado que como se mencionó anteriormente, los servicios con los que este contaría serían dos cuya funcionalidad sería la misma, recibir los archivos enviados por el frontend y realizar la predicción/clasificación de los sentimientos expresados, pero variarían en que uno usaría el modelo entrenado con tweets y el otro el modelo entrenado con los artículos periodísticos. La implementación del *backend* se desarrolló en el lenguaje Python, haciendo uso del *framework* Flask.

Para levantar el backend se deberán descargar las dependencias listadas en el documento “requirements.txt” haciendo uso del comando en la ventana de comandos:

```
$ pip install -U -r requirements.txt
```

Con lo cual se instalarán las librerías requeridas, las cuales son:

- Flask
- Flask-Cors
- Flask-JWT-Extended
- Flask-Migrate
- Flask-RESTful
- Flask-Script
- Flask-SQLAlchemy

- pandas
- torch
- torchvision
- transformers
- jupyter

Y una vez instaladas las dependencias se debe ejecutar el archivo “run.py” con el siguiente comando:

```
$ python run.py
```

Los servicios previamente explicados son de tipo RESTful y estos funcionan de la siguiente manera. En primera instancia se recibe el archivo y este es *parseado* haciendo uso del RequestParser de la librería flask_restful, luego se procederá a leer el archivo y crear un *DataFrame* de la librería *pandas* con él, dado que la manipulación del conjunto de datos se realiza de manera sencilla con este tipo de dato. Una vez generado el *DataFrame*, se procede a limpiar el conjunto de datos, haciendo el preprocesamiento que dio mejores resultados en la experimentación realizada en el segundo objetivo, que consistía en eliminar los caracteres no alfanuméricos, mantener las tildes y convertir todo el texto a minúsculas. Y con ello se procede a *tokenizar* el conjunto de datos, haciendo uso del *BertTokenizer* de la librería *transformers*. Con la información tokenizada se crea el *DataLoader*, de la librería *torch*, para con este tener los inputs correctamente vectorizadas para realizar la inferencia con el modelo. Después se realizan las predicciones. Y finalmente se genera el archivo de salida, el cual contendrá las predicciones realizadas.

Además de los servicios que realizan el análisis de sentimiento a bases de datos textuales se desarrollaron dos servicios que realizan el análisis de sentimiento a textos

independientes, para con ello demostrar de manera visual el funcionamiento de los modelos, por lo cual de igual forma que los que realizan este trabajo de manera masiva, se desarrollaron dos distintos servicios, uno para publicaciones de redes sociales, y el otro para artículos periodísticos, y estos funcionan de igual forma que los mencionados previamente, con la diferencia que estos retornan la clasificación de cada texto, y no todo un archivo con las clasificaciones.

En cuanto a la interfaz web de la aplicación, se implementó un *frontend* en JavaScript, usando la librería de *React* y se hizo uso del *framework* de Material-UI. Esto se realizó por el hecho de que dicho *framework* trabaja de tal manera que sus componentes funcionan de manera *responsive*, generando así una adaptabilidad en la estética de la aplicación, manteniéndola en cualquier tipo de dispositivo.

La interfaz web contará con las tres funcionalidades previamente descritas, la posibilidad de subir lo archivos de texto de tipo “.csv” a los cuales se les realizará el análisis de sentimiento para el caso de artículos periodísticos y para publicaciones de redes sociales, y además se contará con la prueba visual del funcionamiento de los modelos. Está debidamente señalado cada botón asociado a cada funcionalidad con un título por cada uno, donde se indica si es para artículos periodísticos o para publicaciones realizadas en redes sociales, y para el caso de la prueba visual se tiene una sección a parte que será descrita luego. A continuación, se mostrará una captura de la interfaz web implementada.



Figura 8. Interfaz web para pruebas de usuario

Para la parte que involucra la subida de archivos se deberá verificar que estos cuenten con las siguientes condiciones:

- Los de publicaciones de redes sociales deben contar con una columna llamada ‘tweet’, que será donde se encuentra el texto a analizar
- Los de artículos periodísticos deben contar con una columna llamada ‘summary’, que será donde se encuentra el texto a analizar

Para la parte que realiza la prueba visual del funcionamiento de modelos se cuenta con una sección para dicha tarea dentro de la web, la cual en primera instancia permite la selección del tipo de texto el cual ingresará el usuario para posteriormente realizar el análisis de sentimiento sobre este, que será de tipo artículo periodístico o publicación de redes sociales, para que así de esta forma se elija correctamente el modelo que se utilizará. Luego se tiene un campo de texto donde se ingresa el texto a clasificar y finalmente un botón para realizar dicha tarea. Una vez presionado el botón en la parte inferior de dicha sección se mostrará el resultado de dicha clasificación, que podrá ser positivo (P), negativo (N) o neutral (NEU). A continuación, se mostrará una captura del funcionamiento de dicha prueba, donde se ingresó un texto emulando una publicación de alguna red social y se probó el funcionamiento de la prueba.

Prueba del Funcionamiento de los Modelos
 Para el análisis de sentimiento

A continuación se podrá experimentar con textos elegidos por el usuario y así probar la funcionalidad de los modelos implementados. Para ello primero se debe seleccionar el tipo de texto y luego escribir el texto correspondiente.

Artículo Periodístico
 Publicación de Redes Sociales

Ingrese el texto

que feo, no me gusto lo que dijo vizcarra

CLASIFICAR

Clasificación: N

Figura 9. Prueba de funcionamiento de los modelos dentro de la interfaz web

En la captura mostrada se muestra el texto el cual decía: “que feo, no me gusto lo que dijo vizcarra” fue clasificado como negativo, dada que esa es la polaridad del sentimiento expresada en él, con lo cual se probó el correcto funcionamiento de esta.

De esta manera se estaría cumpliendo el segundo objetivo de la presente tesis, dado que se automatiza el proceso de la definición de la polaridad expresada en los textos de artículos periodísticos o de publicaciones realizadas en redes sociales, en este caso referidos a temas políticos peruanos, y se estarían estandarizando los criterios de la definición de este eliminando el sesgo previamente existente dado que ya no se depende del *expertise* de nadie. Definiendo así un pequeño flujo a seguir, el cual consistirá en darle formato de manera manual al archivo de texto según los requerimientos preestablecidos y subirlo a la presente web y darle a la opción de clasificarlo, con lo cual se obtendrá la clasificación de los textos contenidos en este, automatizando la tarea de la definición del sentimiento.

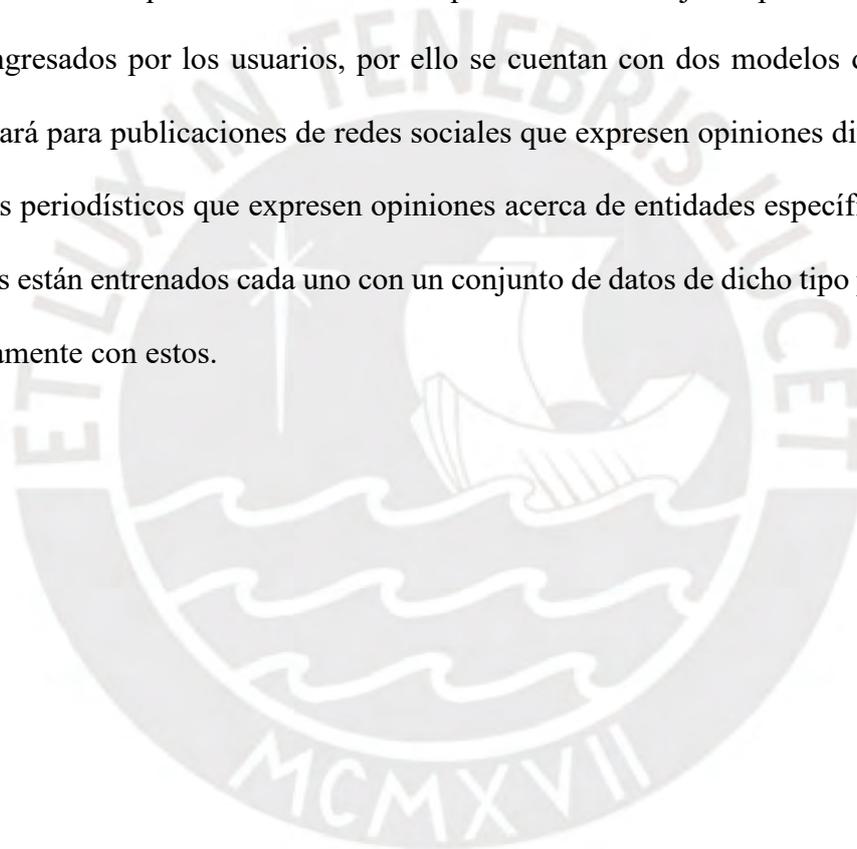
Tanto el código del *frontend* como el del *backend* de la aplicación se encuentran subidos en un gestor de versiones de Git, cuyos enlaces se encuentran detallados en el Anexo J.

5.3 Discusión.

Para poder cumplir el segundo objetivo de la presente tesis que era automatizar la definición de sentimiento de notas periodísticas y estandarizar los criterios usados se implementó una aplicación web que pueda cumplir ello, por lo cual esta se implementó con dos funcionalidades principales que cumplan la automatización de dicho proceso, lo cual se hizo de la siguiente manera. En primer lugar, se permite la subida de archivos mediante una interfaz realizada para el usuario, el cual sólo deberá subir un archivo con el formato requerido para cada caso, y este se enviará al *backend* de la aplicación, donde haciendo uso de los modelos optimizados obtenidos en el último resultado del objetivo dos, se definirán

los sentimientos expresados en cada texto, y con ello se generará un archivo de salida el cual cuenta con la clasificación de la polaridad del sentimiento expresada en el texto. Y además se desarrolló una pequeña funcionalidad que servirá para demostrar el funcionamiento de los modelos en la interfaz de manera visual, donde se permite clasificar textos ingresados directamente por el usuario.

Cabe destacar que el funcionamiento óptimo de estos flujos dependerá de los textos que serán ingresados por los usuarios, por ello se cuentan con dos modelos distintos, uno que funcionará para publicaciones de redes sociales que expresen opiniones diversas y otro para artículos periodísticos que expresen opiniones acerca de entidades específicas, dado que los modelos están entrenados cada uno con un conjunto de datos de dicho tipo para así funcionar correctamente con estos.



Capítulo 6. Dominio de información periodística estructurado y clasificado

6.1 Introducción.

En el presente capítulo de la redacción de resultados se describirá detalladamente los pasos realizados para la obtención de estos, y así poder cumplir el último objetivo de la presente tesis, el cual consiste en estructurar un dominio de información periodística que provendrá de artículos periodísticos y de tweets y con ellos poder realizar un análisis de sentimiento, teniendo así la clasificación de la polaridad de los sentimientos expresados en dichos textos.

El dominio escogido a tratar será: noticias, artículos y tweets que tratan temas de política peruana ocurridos en el año 2020. Por ello en el siguiente inciso se describirá cómo se extrajo la información de internet para así armar una base de datos perteneciente al dominio establecido, y cómo se estructuró la misma para así finalmente subir estas bases de datos textuales como archivos a la interfaz web realizada en el segundo objetivo y así probar el flujo a seguir y verificar el funcionamiento de los modelos desarrollados.

6.2 Resultados Alcanzados.

6.2.1 Base de datos textual de artículos periodísticos.

Este primer resultado consiste en una base de datos textual compuesta por artículos periodísticos pertenecientes al dominio previamente escogido, el cual es temas de política peruana del 2020, la cual luego se procederá a estructurar para luego usar esta como caso de estudio de la presente tesis y con esta probar el modelo algorítmico desarrollado.

En primera instancia, para generar la base textual de artículos periodísticos con la cual se evaluará el modelo se realizó la tarea de *web scraping*, que consiste en el desarrollo de unos

programas que extraigan la información deseada de sitios web determinados, de las páginas web de los diarios peruanos Gestión y El Comercio, de los cuales se extrajeron un total de 3 323 artículos referidos a política respectivamente. Para realizar la tarea de *web scraping* se utilizaron las librerías *scrapy* y *bs4* pertenecientes al lenguaje Python, y para juntar lo obtenido de ambos diarios y formatearlo correctamente se usó la librería *panda*, con la cual se generó un archivo de tipo “.csv” con la base de datos resultante

Los archivos generados están compuestos por un conjunto de artículos periodísticos que tienen la siguiente información por cada uno de ellos:

- Título
- Subtítulo
- Contenido

Con ello se cumplieron los indicadores de verificación establecidos, dado que se exigía un mínimo de 100 artículos y finalmente se tiene un total de 3 323 artículos.

Dicho archivo generado, que compone la base de datos textual de artículos periodísticos, se encuentra en un repositorio de Google Drive, cuya ubicación está detallada en el Anexo K, el cual tiene como nombre “BDNoticias.csv”

6.2.2 Base de datos textual de tweets.

El segundo resultado consiste en una base de datos textual compuesta por tweets pertenecientes al dominio previamente escogido, el cual es temas de política peruana del 2020, la cual luego se procederá a estructurar para luego usar esta como caso de estudio de la presente tesis y con esta probar el modelo algorítmico desarrollado.

Para la obtención de la base de datos textual de tweets se hizo uso de la propia API de Twitter, para lo cual se hizo uso de la librería *tweepy* perteneciente al lenguaje Python, donde directamente se hizo una consulta a Twitter para obtener un conjunto de tweets referidos a política peruana, esta consulta utilizó las palabras clave “Perú” y “Política”, además de ello se delimitó la consulta a tweets realizados en el año 2020.

El archivo generado está compuesto por un conjunto de tweets y tiene los siguientes campos por cada uno de estos:

- ID Tweet
- Fecha de publicación
- Contenido del Tweet

Con ello se generó un archivo de tipo “.csv” con un total de 10 000 tweets, cumpliendo con ello el indicador objetivamente variable establecido, el cual exigía un mínimo de 1 000 tweets.

Este archivo generado compondrá la base de datos textual de tweets y se encuentra en un repositorio de Google Drive, cuya ubicación está detallada en el Anexo K, y este tiene como nombre “BDTweets.csv”.

6.2.3 Base de datos textual estructurada y clasificada de la información recolectada .

El tercer resultado consiste en una base de datos textual debidamente estructurada de tal forma que esta pueda servir como información de entrada para interfaz web desarrollada en el segundo objetivo y así poder realizar un análisis sentimental sobre esta base de datos. Para ello se utilizarán las bases de datos previamente generadas. Cabe destacar que estas se estructurarán de manera independiente, dado que se entrenaron dos instancias distintas del

modelo, será una exclusivamente para publicaciones de redes sociales, y otra para artículos periodísticos que hablan de alguna entidad en sí, esto dadas las diferencias entre estos dos tipos de textos y la composición de estos.

La estructuración de la información recolectada se realizó haciendo uso de la librería pandas de Python, dado que esta permite fácilmente manipular archivo de este tipo, y se realizó de la siguiente manera:

- **Estructuración de base de datos textual de artículos periodísticos:**

En primer lugar, se debe especificar que el análisis sentimental de las noticias se realizará en base a un resumen de la noticia, el cuál en el caso del conjunto de datos extraído en el resultado previo de diarios peruanos se generará concatenando el título de la noticia, con el subtítulo de esta.

Esto se hará dado que el análisis sentimental a realizar sobre artículos periodísticos tomará como base el proceso usado por la empresa de monitoreo de medios IP noticias, donde la clasificación de la polaridad del sentimiento expresada en los artículos se hace en base a un resumen con el que ellos ya cuentan

Por ende, finalmente la base de datos de artículos periodísticos quedará como se muestra en la imagen a continuación.

| | subtitle | title | summary |
|---|--|--|--|
| 0 | El presidente Martín Vizcarra defendió la des... | Vizcarra sobre designación de Zeballos: "El v... | Vizcarra sobre designación de Zeballos: "El v... |
| 1 | El primer vicepresidente del Congreso indicó ... | Congresista Valdez: Ministro del Interior deb... | Congresista Valdez: Ministro del Interior deb... |
| 2 | La legisladora de Fuerza Popular Martha Cháve... | Martha Chávez: Vicente Zeballos "debió ir a B... | Martha Chávez: Vicente Zeballos "debió ir a B... |
| 3 | La legisladora consideró que el presidente VI... | Martha Chávez tras críticas de Vizcarra: "Los... | Martha Chávez tras críticas de Vizcarra: "Los... |
| 4 | La titular de Justicia, Ana Neyra, detalló lo... | Minjus dice que no se debería promover el asc... | Minjus dice que no se debería promover el asc... |

Figura 10. Base de datos textual de artículos periodísticos estructurada

El archivo generado finalmente se encuentra en el repositorio de Google Drive, cuya ubicación está detallada en el Anexo K, y este tiene como nombre “BDNoticias Estructurada.csv”.

- **Estructuración de base de datos textual de tweets:**

El análisis sentimental de los tweets se realizará por tweet, por lo que se usarán dos columnas, una será la del identificador del tweet y el otro contenido del tweet, como se muestra en la imagen a continuación.

| | id | tweet |
|---|---------------------|---|
| 0 | 1304184807941955584 | RT @JuanSheput: Los teóricos señalan que la va... |
| 1 | 1304184794356613120 | RT @gvillasis: Los Cuellos Blancos del Puerto:... |
| 2 | 1304184722046816256 | RT @RudyRiveraB: La MAFIA DE LA CORRUPCIÓN |
| 3 | 1304184713435844610 | Me estresa la inestabilidad política del Perú... |
| 4 | 1304184706389495812 | RT @SAYSAYSAYSA: @GeneralDelAire1 seria incons... |

Figura 11. Base de datos textual de tweets estructurada

El archivo generado finalmente se encuentra en el repositorio de Google Drive, cuya ubicación está detallada en el Anexo K, y este tiene como nombre “BDTweetsEstructurada.csv”.

Y finalmente para probar la interfaz implementada, el flujo desarrollado y cumplir con el tercer objetivo de la presente tesis, se debe clasificar los textos extraídos y estructurados previamente.

Para ello se subió cada archivo, uno por uno a la interfaz y se realizó la clasificación. Con lo que se obtuvieron los siguientes resultados:

- En la base de datos textual de artículos periodísticos se obtuvieron:
 - 2 951 artículos que expresaban sentimientos neutros

- 285 artículos que expresaban sentimientos negativos
- 87 artículos que expresaban sentimientos positivos

A continuación, se mostrará una imagen de un fragmento del archivo resultante luego de la ejecución del procedimiento definido.

| | subtitle | title | summary | prediction |
|---|--|---|---|------------|
| 0 | El presidente Martín Vizcarra defendió la d... | Vizcarra sobre designación de Zeballos: â□□E... | vizcarra sobre designación de zeballos â el v... | Neutro |
| 1 | El primer vicepresidente del Congreso indic... | Congresista Valdez: Ministro del Interior deb... | congresista valdez ministro del interior deber... | Neutro |
| 2 | La legisladora de Fuerza Popular Martha ChÃ¼v... | Martha ChÃ¼vez: Vicente Zeballos â□□debiÃ³ ir... | martha chÃ¼ vez vicente zeballos â debiÃ¡ ir a b... | Neutro |
| 3 | La legisladora considerÃ³ que el presidente V... | Martha ChÃ¼vez tras crÃ¡ticas de Vizcarra: â□□... | martha chÃ¼ vez tras crÃ¡ticas de vizcarra â lo... | Neutro |
| 4 | La titular de Justicia, Ana Neyra, detallÃ³ l... | Minjus dice que no se deberÃ¡a promover el as... | minjus dice que no se deberÃ¡ a promover el asc... | Neutro |

Figura 12. Archivo resultante del análisis de sentimiento de artículos periodísticos

El tiempo de ejecución aproximado de la clasificación de dicha base de datos fue de 7 minutos aproximadamente.

- En la base de datos textual de tweets s se obtuvieron:
 - 3 029 tweets que expresaban sentimientos neutros
 - 5 917 tweets que expresaban sentimientos negativos
 - 1 054 tweets que expresaban sentimientos positivos

A continuación, se mostrará una imagen de un fragmento del archivo resultante luego de la ejecución del procedimiento definido.

| | id | tweet | prediction |
|---|---------------------|---|------------|
| 0 | 1304184807941955584 | rt juansheput los teóricos señalan que la vaca... | N |
| 1 | 1304184794356613120 | rt gvillasis los cuellos blancos del puerto el... | NEU |
| 2 | 1304184722046816256 | rt rudyriverab la mafia de la corrupción | NEU |
| 3 | 1304184713435844610 | me estresa la inestabilidad politica del Perú ... | NEU |
| 4 | 1304184706389495812 | rt saysaysaysa generaldelaire1 seria inconstit... | NEU |

Figura 13. Archivo resultante del análisis de sentimiento de tweets

El tiempo de ejecución aproximado de la clasificación de dicha base de datos fue de 15 minutos aproximadamente.

La ejecución de cada una de estas predicciones sirvió de prueba unitaria por cada funcionalidad disponible de la aplicación desarrollada, con lo cual se tiene un 100% de funcionamiento de estas.

6.3 Discusión.

Para poder cumplir el último objetivo de la tesis planteada, que consistía en estructurar un dominio de información periodística para realizar un análisis de sentimiento, se definió como dominio para el caso de estudio planteado un conjunto de noticias extraídas de las webs del diario Gestión y El Comercio, y un conjunto de tweets referidos a hechos de política peruana ocurridos en el año 2020, los cuales fueron extraídos apropiadamente, y cabe destacar que se extrajo 10 veces más la cantidad previamente establecida para poder con ello trabajar esta conjunto de datos y tener un margen de error de información que pueda resultar inútil. Luego estos fueron estructurados, para con ello finalmente tener dos bases de datos estructuradas. Una base de datos perteneciente al dominio establecido será la de noticias, la cual será clasificada en base a un resumen de esta generado en base a la concatenación de este con el subtítulo de esta. Y la segunda base de datos será la de tweets, la cual contiene un identificador por cada tweet y el contenido correspondiente de estos.

Y finalmente con estas bases de datos estructuras se realizó el análisis de sentimiento como caso de prueba de la presente tesis, lo cual se realizó a través del uso de la aplicación web desarrollada para el segundo objetivo de la presente tesis, donde se subieron los presentes archivos, cada uno de manera independiente y se realizó dicha la clasificación de la polaridad de los sentimientos expresados en los textos.

Así como en este caso se utilizaron textos del dominio político se podría escoger otro dominio y el modelo funcionaría correctamente, pero en caso se desee que funcione de manera específica a un dominio se debería entrenar exclusivamente el modelo del primer objetivo con el conjunto de datos perteneciente a este, con un riesgo a que este esté sobreentrenado y solo funcione para dicho rubro, por ello en este caso se hizo entrenaron los modelos con un conjunto de datos más diverso para que este pueda aprender los patrones de diversos textos y no se sesgue en un solo dominio.



Capítulo 7. Conclusiones y trabajos futuros.

7.1 Conclusiones.

Según lo investigado previamente al desarrollo de la presente tesis se observó la existencia de procedimientos ambiguos para lo que es la realización del análisis de sentimiento y sobre todo para textos en español y para el caso de estudio escogido que es específicamente dicho análisis en notas periodísticas y redes sociales, lo cual se daba por el hecho de la escasa cantidad de herramientas existentes para la presente tarea, lo cual hacía que dicho proceso de clasificación de las polaridades de los sentimientos expresadas en los textos se realice de manera manual por expertos en el tema, generándose así resultados ambiguos y sesgados según la experiencia del encargado, lo cual generaba resultados que no eran del todo fiables restándole valor a dicha tarea, además del hecho de que realizarlo de manera totalmente manual resultaba muy pesado y se realizaba en un periodo largo de tiempo.

Por ello se determinó como principal objetivo el implementar un modelo algorítmico de análisis de sentimiento en notas periodísticas y redes sociales, con lo cual se contaría con una solución computacional que serviría para automatizar la tarea de la definición de la polaridad de sentimientos expresados en dichos textos haciendo uso de técnicas de aprendizaje profundo. Partiendo de esa premisa se desarrollaron tres objetivos específicos.

En primera instancia se tenía como primer objetivo específico el implementar un modelo que desarrolle análisis sentimental para un conjunto de datos en español, para lo cual se dividió dicha tarea en cuatro partes. Primero se escogió una arquitectura base sobre la cual se iba a desarrollar el modelo, donde se escogió como arquitectura base al modelo pre-entrenado BERT, el cual se entrenó mediante *fine tuning* y así haciendo uso de *transfer* y

ajustando el modelo para que funcione específicamente para la tarea de análisis de sentimiento, como este era el modelo base y solo se quería comprobar el correcto funcionamiento de este para el análisis de sentimiento este se probó con un conjunto de tweets en inglés, dado que BERT está pre entrenado con textos en dicho idioma y se obtuvo un score de precisión de este primer modelo de 86,55% lo cual era bastante alto por el hecho de que BERT está sumamente optimizado para trabajar con textos en inglés. Para la segunda parte se debía adaptar la arquitectura base para funcione correctamente con textos en español, por lo cual se hizo el cambio de BERT a BETO, el cual es también un modelo pre entrenado con las mismas características que BERT pero con textos en español, por lo cual se realizó dicho cambio, se ajustó el modelo y se entrenó con un conjunto de tweets en español, con lo que se obtuvo un score de precisión de 68,55%, lo cual debía irse mejorando. Por ello como tercer paso se procedió a implementar dos modelos para la tarea de análisis de sentimiento, uno por cada tipo de textos planteado en la tesis, uno era para publicaciones de redes sociales, y otro para artículos periodísticos. Para ir mejorando el modelo se experimentó con este, con las variaciones existentes de BETO, con diversos preprocesamientos de textos y finalmente se determinó que la mejor combinación era BETO *uncased*, el cual es un modelo pre entrenado de tal manera que este no hace distinción entre mayúsculas y minúsculas de los textos, y un preprocesamiento del texto donde se eliminan los caracteres no alfanuméricos pero si se mantienen las tildes, con lo cual se entrenaron los dos modelos desarrollados y así el de artículos periodísticos obtuvo un score de precisión de 85.24% y el de publicaciones de redes sociales tuvo un score de precisión de 75.98%, resultados bastante buenos, donde si bien el de artículos periodísticos es bastante mayor se da debido a la que se entrenó con un volumen mayor de información que el de redes sociales. Finalmente, como cuarto paso se procedió a experimentar con la optimización de los modelos para mejorar este score de precisión, donde

finalmente se determinó que utilizaría *damW* con ajuste de coseno con calentamiento (*cosine_schedule_with_warmup*) y para artículos periodísticos a *Adafactor* con ajuste de coseno con reinicios forzados y calentamiento (*cosine with hard restarts schedule with warmup*), teniendo así finalmente un score de precisión de 76.42% en el modelo de publicaciones en redes sociales y un 88.39% en el de artículos periodísticos, cumpliendo así el objetivo específico planteado.

Como segundo objetivo se tenía la automatización de la definición de sentimiento de notas periodísticas y con ello estandarizar los criterios usados, para lo cual se decidió implementar una aplicación web mediante la cual se automatice dicha tarea y no se deba realizar de manera manual, eliminando así el sesgo de la experiencia del clasificador y estandarizando así el criterio de clasificación. La interfaz web de la aplicación implementada se desarrolló de tal forma que se brinda una plataforma donde el usuario podrá en primera instancia ver el funcionamiento de los modelos a través de una sección donde se permite el ingreso de un texto a clasificar, y como principal funcionalidad para poder cumplir con el objetivo planteado se le permite al usuario el subir archivos, los cuales deberán seguir un formato detallado en el quinto capítulo de la presente tesis, que contengan textos a los cuales se les desea realizar el análisis de sentimiento y se realice esta tarea de manera automatizada. Existe un componente para la subida de archivos de textos que son publicaciones de redes sociales y otro para los artículos periodísticos, esto por el hecho de que se manejan dos modelos distintos. La aplicación implementada tiene cargados los modelos previamente implementados y realiza hace uso de ellos para realizar la clasificación de los textos. De esta manera se automatiza la tarea de definición de sentimiento de artículos periodísticos y de publicaciones de redes sociales cumpliendo así el segundo objetivo planteado, y para ver mejor el cumplimiento de este objetivo se planteó un tercer objetivo el cual consistía en

estructurar un dominio de información periodística para realizar un análisis de sentimiento, para lo cual se hará uso de la aplicación previamente descrita y servirá como caso de estudio para probar el correcto funcionamiento de la tesis y un flujo más real de cómo se puede aplicar lo implementado a la vida cotidiana. Por ello se empezó con definir el dominio a usar, y se eligió la política peruana del 2020. Con ello se procedió a extraer un conjunto de tweets de la web, haciendo uso de la API de Twitter y se realizó *web scrapping* para extraer noticias de El Comercio y del Diario Gestión, con lo que se formaron dos bases de datos textuales, una por cada tipo de texto manejado. Con ello se estructuró dichas bases de datos siguiendo el formato requerido y se subió de manera independiente cada archivo textual a la aplicación desarrollada, con lo cual finalmente se obtuvo satisfactoriamente dos archivos finales con la clasificación de la polaridad de los sentimientos expresados en cada texto que compone cada base de datos comprobando así el funcionamiento correcto de dicho flujo planteado en el segundo objetivo y además cumpliendo el tercer objetivo de la tesis.

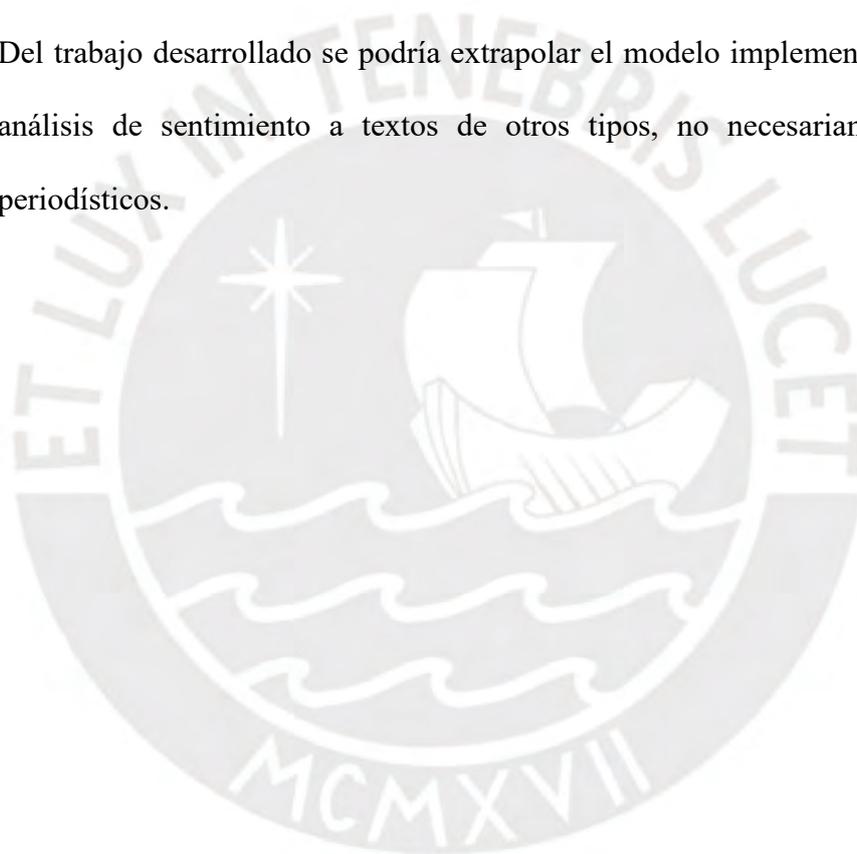
Finalmente habiendo cumplido los tres objetivos específicos de la tesis se cumplió también el objetivo principal de la tesis al implementar un modelo algorítmico de análisis de sentimiento en notas periodísticas y redes sociales de manera satisfactoria.

7.2 Trabajos futuros.

- La idea del desarrollo de la presente tesis surgió de la problemática surgida en una empresa dedicada al monitoreo de medios, donde una de las tareas realizadas consistía en el análisis de sentimiento orientado a opiniones expresadas en base a entidades en concreto, por ello también se desarrolló otro trabajo de tesis dedicado al reconocimiento de entidades mencionadas en textos, por ello como trabajo a futuro está la integración con dicho trabajo de tesis de tal manera que las salidas del modelo

implementado en dicho trabajo sirvan como entrada para el modelo desarrollado en la presente tesis, solucionando así completamente la problemática global existente.

- Del trabajo desarrollado se podría extrapolar el modelo implementado para realizar la clasificación de textos en español, pero ya no orientado a la polaridad de los sentimientos expresados, sino a otros tipos de clasificación como lo podría ser por tipo de texto o por el contenido del texto.
- Del trabajo desarrollado se podría extrapolar el modelo implementado para realizar análisis de sentimiento a textos de otros tipos, no necesariamente de medios periodísticos.



Referencias

Ait Hammou, B., Ait Lahcen, A. and Mouline, S. (2020) 'Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics', *Information Processing and Management*. doi: 10.1016/j.ipm.2019.102122.

Dong, Y. *et al.* (2020) 'A Fusion Model-Based Label Embedding and Self-Interaction Attention for Text Classification', *IEEE Access*, 8, pp. 30548–30559. doi: 10.1109/ACCESS.2019.2954985.

Lien Minh, D. *et al.* (2018) 'Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network', *IEEE Access*. doi: 10.1109/ACCESS.2018.2868970.

Liu, S. *et al.* (2019) 'R-Trans: RNN Transformer Network for Chinese Machine Reading Comprehension', *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 7, pp. 27736–27745. doi: 10.1109/ACCESS.2019.2901547.

Rida-E-Fatima, S. *et al.* (2019) 'A Multi-Layer Dual Attention Deep Learning Model With Refined Word Embeddings for Aspect-Based Sentiment Analysis', *IEEE Access*. doi: 10.1109/access.2019.2927281.

Shao, T. *et al.* (2019) 'Transformer-Based Neural Network for Answer Selection in Question Answering', *IEEE Access*, 7, pp. 26146–26156. doi: 10.1109/ACCESS.2019.2900753.

Sun, X. *et al.* (2019) 'Emotional conversation generation based on a Bayesian deep neural network', *ACM Transactions on Information Systems*, 38(1). doi: 10.1145/3368960.

Tao, Jie and Fang, X. (2020) 'Toward multi-label sentiment analysis: a transfer learning based approach', *Journal of Big Data*. doi: 10.1186/s40537-019-0278-0.

Tao, J. and Fang, X. (2020) 'Toward multi-label sentiment analysis: a transfer learning based approach', *Journal of Big Data*, 7(1). doi: 10.1186/s40537-019-0278-0.

Vaswani, A. *et al.* (2017) 'Attention is all you need', in *Advances in Neural Information Processing Systems*.

Yin, H. *et al.* (2019) 'Capsule Network with Identifying Transferable Knowledge for Cross-Domain Sentiment Classification', *IEEE Access*. doi: 10.1109/ACCESS.2019.2948628.

Yolchuyeva, S., Németh, G. and Gyires-Tóth, B. (2019) 'Self-attention networks for intent detection', in *International Conference Recent Advances in Natural Language Processing, RANLP*, pp. 1373–1379. doi: 10.26615/978-954-452-056-4_157.

Zhang, L. and Liu, B. (2016) 'Sentiment Analysis and Opinion Mining', in *Encyclopedia of Machine Learning and Data Mining*. doi: 10.1007/978-1-4899-7502-7_907-1.

Zheng, J. and Zheng, L. (2019) 'A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification', *IEEE Access*, 7, pp. 106673–106685. doi: 10.1109/ACCESS.2019.2932619.

Cortez, A., Vega, H. and Pariona, J. (2011) 'Procesamiento de lenguaje natural robusto', *Primer encuentro de GruPos de investigación sobre Procesamiento del lenguaje*, 2013(3), p. 147.

Jurafsky, D. and Martin, J. (2002) 'Speech and Language Processing. An Introduction to Natural Language

Processing, Computational Linguistics, and Speech Recognition', *Zeitschrift für Sprachwissenschaft*, 21(1), pp. 134–135. doi: 10.1515/zfsw.2002.21.1.134.

Jiang, D. *et al.* (2017) 'Sentiment computing for the news event based on the social media big data', *IEEE Access*. doi: 10.1109/ACCESS.2016.2607218.

Zhang, S. *et al.* (2015) 'Code Consistent Hashing Based on Information-Theoretic Criterion', *IEEE Transactions on Big Data*. doi: 10.1109/tbdata.2015.2499191.

Bekkers, V., Edwards, A., & de Kool, D. (2013). Social media monitoring: Responsive governance in the shadow of surveillance?. *Government Information Quarterly*, 30(4), 335-342.

Zhang, B., & Vos, M. (2014). Social media monitoring: aims, methods, and challenges for international companies. *Corporate Communications: An International Journal*.

Batrinca, B., & Treleaven, P. C. (2015). Social media analytics: a survey of techniques, tools and platforms. *Ai & Society*, 30(1), 89-116.

Ebrahimi, M., Yazdavar, A. H., & Sheth, A. (2017). Challenges of sentiment analysis for dynamic events. *IEEE Intelligent Systems*, 32(5), 70-75.

Younis, E. M. (2015). Sentiment analysis and text mining for social media microblogs using open source tools: an empirical study. *International Journal of Computer Applications*, 112(5).

Deng, L., & Liu, Y. (Eds.). (2018). *Deep learning in natural language processing*. Springer.

McNamara, D. S., Allen, L. K., Crossley, S. A., Dascalu, M., & Perret, C. A. (2017). Natural language processing and learning analytics. *Handbook of learning analytics*, 93-104.

Pla, F., & Hurtado, L. F. (2018). Spanish sentiment analysis in Twitter at the TASS workshop. *Language Resources and Evaluation*, 52(2), 645-672.

del Arco, F. M. P., Valdivia, M. T. M., Zafra, S. M. J., González, M. D. M., & Cámara, E. M. (2016). COPOS: corpus of patient opinions in spanish. application of sentiment analysis techniques. *Procesamiento del Lenguaje Natural*, 57, 83-90.

Díaz-Galiano, M. C., García-Vega, M., Casasola, E., Chiruzzo, L., García-Cumbreras, M. Á., Cámara, E. M., ... & Graff, M. (2019). Overview of TASS 2019: One More Further for the Global Spanish Sentiment Analysis Corpus*.

Paredes-Valverde, M. A., Colomo-Palacios, R., Salas-Zárate, M. D. P., & Valencia-García, R. (2017). Sentiment analysis in Spanish for improvement of products and services: A deep learning approach. *Scientific Programming*, 2017.

Tellez, E. S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Siordia, O. S., & Villaseñor, E. A. (2017). A case study of Spanish text transformations for twitter sentiment analysis. *Expert Systems with Applications*, 81, 457-471.

Flask (n.d.) *Welcome to Flask — Flask Documentation (1.1.x)*. Available at: <https://flask.palletsprojects.com/en/1.1.x/> (Accessed: 21 June 2020).

Git (n.d.) *Git*. Available at: <https://git-scm.com/> (Accessed: 21 June 2020).

De Ita, M. C. (2019) *La transformación digital llegó a los medios de comunicación*,

Conexión ESAN. Available at:

<https://www.esan.edu.pe/conexion/actualidad/2019/07/17/la-transformacion-digital-llego-a-los-medios-de-comunicacion/> (Accessed: 21 June 2020).

Mozilla (n.d.) *JavaScript | MDN, MDN web docs*. Available at:

<https://developer.mozilla.org/es/docs/Web/JavaScript> (Accessed: 21 June 2020).

Nimshi, V. and Konam, S. (2020) *Abridging clinical conversations using Python | Our Success Stories | Python.org*. Available at: <https://www.python.org/success-stories/abridging-clinical-conversations-using-python/> (Accessed: 21 June 2020).

pandas (n.d.) *pandas - Python Data Analysis Library*. Available at: <https://pandas.pydata.org/> (Accessed: 21 June 2020).

Project Jupyter (no date) *Project Jupyter*. Available at: <https://jupyter.org/index.html> (Accessed: 21 June 2020).

Python (n.d.) *Welcome to Python.org*. Available at: <https://www.python.org/> (Accessed: 21 June 2020).

PyTorch (n.d.) *PyTorch*. Available at: <https://pytorch.org/> (Accessed: 21 June 2020).

React (n.d.) *React*. Available at: <https://es.reactjs.org/> (Accessed: 21 June 2020).

Venegas, R. (2007) 'Clasificación de textos académicos en función de su contenido léxico-semántico', *Revista Signos*, 40(63), pp. 239–271. doi: 10.4067/s0718-09342007000100012.

Estudia Perú (2020) *Ingeniería de Sistemas | Mejores Universidades | Campo Laboral y Sueldo*. Available at: <https://estudiaperu.pe/carrera-pregrado/ingenieria-de-sistemas/> (Accessed: 8 July 2020).

huggingface (2020) *Transformers — transformers 3.1.0 documentation*. Available at: <https://huggingface.co/transformers/> (Accessed: 11 September 2020).

Wang, B. *et al.* (2016) *SMILE Twitter Emotion dataset*. Available at: https://figshare.com/articles/dataset/smile_annotations_final_csv/3187909/2 (Accessed: 11 September 2020).

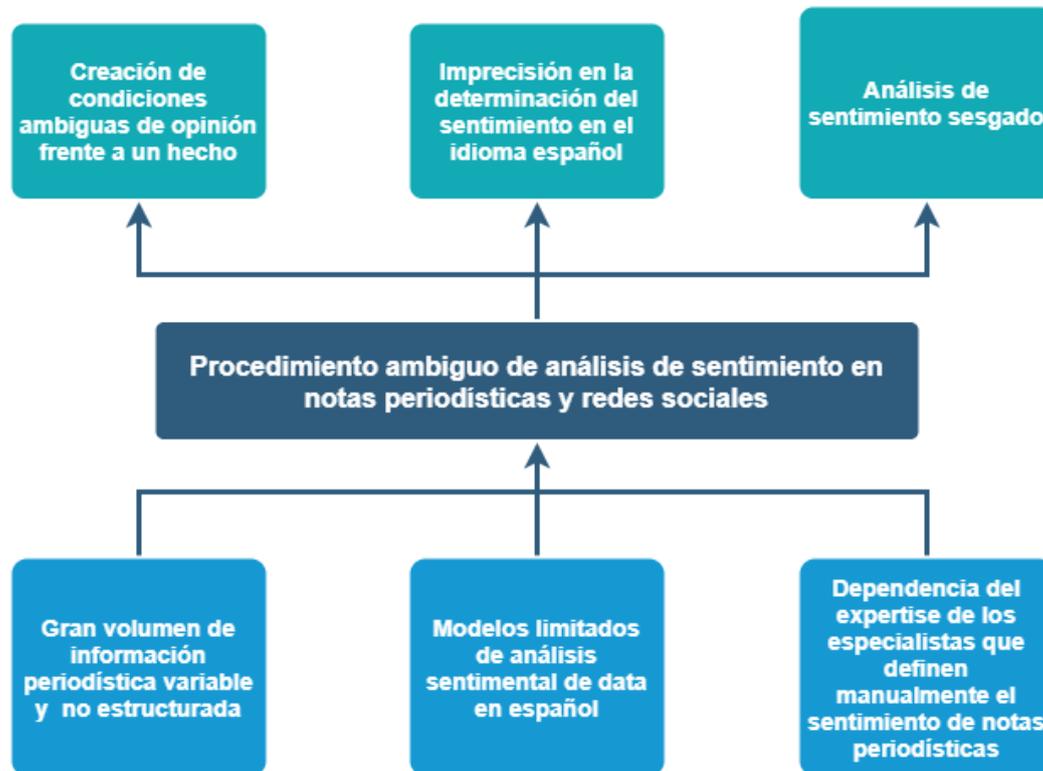
Departamento de Ciencias de la Computación Universidad de Chile (2020) *BETO - Spanish version of the BERT model*. Available at: <https://github.com/dccuchile/beto> (Accessed: 11 September 2020).

Deguchi, H., Tamura, A., & Ninomiya, T. (2019). Dependency-based self-attention for transformer NMT. *International Conference Recent Advances in Natural Language Processing, RANLP, 2019-Septe*, 239–246. https://doi.org/10.26615/978-954-452-056-4_028

Zhang, Q., & Lu, R. (2019). A Multi-Attention Network for aspect-level sentiment analysis. *Future Internet*, 11(7). <https://doi.org/10.3390/fi11070157>

Anexos

Anexo A: Árbol de problemas



Anexo B: Formulario de extracción

El anexo es un documento externo en Excel el cual contiene dicho formulario lleno, con el siguiente nombre “**Anexo B: Formulario de extracción.xlsx**” y se encuentra en el siguiente enlace:

https://docs.google.com/spreadsheets/d/1Vaw6Og8oQ7YBji_dvWady2zx8-DaqQtewzNV6Yzodz0/edit?usp=sharing

Anexo C: Repositorio de Modelos Computacionales Desarrollados

El anexo es la ruta de una carpeta pública de Google Drive, en la cual se encuentra el código fuente de los modelos computacionales implementados para poder cumplir el segundo objetivo de la presente tesis. El enlace es el siguiente:

https://drive.google.com/drive/folders/14ODW-u7XChnuCUK54Rs5LUwBITxCQ_2d?usp=sharing

Anexo D: Reporte de Arquitectura de Modelo Base

| Reporte de Modelo Base | | | |
|--|--|--------------|-------------------------------|
| Información del modelo | | | |
| Nombre del modelo | bert-base-uncased | | |
| Idioma | Inglés | | |
| Tipo | Uncased | | |
| Dataset usado | | | |
| Nombre | smileannotationsfinal.csv | | |
| Contenido | Tweets | | |
| Idioma | Inglés | | |
| Longitud | 3085 | | |
| Sentimientos expresados | happy, angry, disgust, sad, surprise, non-relevant | | |
| Comentarios | Data sin limpiar | | |
| Resultados de entrenamiento | | | |
| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
| 1 | 0,7342 | 0,8465 | 69 |
| 2 | 0,7266 | 0,8393 | 69 |
| 3 | 0,7738 | 0,8506 | 69 |
| 4 | 0,7663 | 0,8604 | 69 |
| 5 | 0,8291 | 0,8608 | 68 |
| 6 | 0,8282 | 0,8641 | 68 |
| 7 | 0,8189 | 0,8645 | 68 |
| 8 | 0,8239 | 0,8595 | 68 |
| 9 | 0,8198 | 0,8595 | 68 |
| 10 | 0,8217 | 0,8595 | 68 |
| | | Total | 684 |
| Modelo elegido para la evaluación | | | |
| Modelo elegido para la evaluación | BERT_ft_epoch7.model | | |
| Resultados de evaluación | | | |
| Sentimiento | Aciertos | Total | Porcentaje |
| happy | 162 | 171 | 94,74% |
| non-relevant | 20 | 32 | 62,50% |
| angry | 6 | 9 | 66,67% |
| disgust | 0 | 1 | 0,00% |
| sad | 3 | 5 | 60,00% |
| surprise | 2 | 5 | 40,00% |
| Total | 193 | 223 | 86,55% |

Anexo E: Reporte de entrenamiento de Modelo Base en español

| Reporte de Modelo Base en Español | | | |
|--|---|--|-------------------------------|
| Información del modelo | | | |
| Nombre del modelo | bert-base-spanish-wwm-cased | | |
| Idioma | Español | | |
| Tipo | Cased | | |
| Dataset usado | | | |
| Nombre | TASS2019_country_XX_train.txt* | *Se juntaron los datasets de CR, ES, PE y UY | |
| Contenido | Tweets | | |
| Idioma | Español | | |
| Longitud | 3811 | | |
| Sentimientos expresados | P (Positivo), N (Negativo), NEU (Neutral) | | |
| Comentarios | Data sin limpiar | | |
| Resultados de entrenamiento | | | |
| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
| 1 | 0,8045 | 0,6084 | 23 |
| 2 | 0,7450 | 0,6314 | 23 |
| 3 | 0,7274 | 0,6908 | 23 |
| 4 | 0,7549 | 0,7050 | 23 |
| 5 | 0,8009 | 0,6980 | 23 |
| 6 | 0,9014 | 0,6834 | 23 |
| 7 | 1,0317 | 0,7095 | 23 |
| 8 | 1,1175 | 0,7003 | 23 |
| 9 | 1,3364 | 0,6884 | 23 |
| 10 | 1,4019 | 0,6838 | 23 |
| 11 | 1,3875 | 0,7014 | 23 |
| 12 | 1,4891 | 0,7037 | 23 |
| 13 | 1,5609 | 0,7028 | 23 |
| 14 | 1,6164 | 0,7041 | 23 |
| 15 | 1,6437 | 0,7055 | 23 |
| 16 | 1,7351 | 0,6998 | 23 |
| 17 | 1,8190 | 0,6837 | 23 |
| 18 | 1,7795 | 0,7018 | 23 |
| 19 | 1,8017 | 0,6975 | 23 |
| 20 | 1,8144 | 0,7029 | 23 |
| | | Total | 460 |
| Modelo elegido para la evaluación | | | |
| Modelo elegido para la evaluación | Bert_R_epoch7_model | | |
| Resultados de evaluación | | | |
| Sentimiento | Aciertos | Total | Porcentaje |
| P | 341 | 443 | 76,96% |
| N | 407 | 534 | 76,22% |
| NEU | 114 | 247 | 46,15% |
| Total | 862 | 1224 | 70,42% |

Anexo F: Reporte de entrenamiento del modelo para tweets en español

| Reporte de Modelo en Español para Tweets | | | |
|--|--|--|-------------------------------|
| Información del modelo | | | |
| Nombre del modelo | bert-base-spanish-wwm-uncased | | |
| Idioma | Español | | |
| Tipo | Uncased | | |
| Dataset usado | | | |
| Nombre | TASS2019_country_XX_train.xml* | *Se juntaron los datasets de CR, ES, PE y UY | |
| Contenido | Tweets | | |
| Idioma | Español | | |
| Longitud | 3811 | | |
| Sentimientos expresados | P (Positivo), N (Negativo), NEU (Neutral) | | |
| Comentarios | Se eliminaron los caracteres no alfanuméricos, se mantuvieron las tildes | | |
| Resultados de entrenamiento | | | |
| N° Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
| 1 | 0,7409 | 0,6216 | 21 |
| 2 | 0,6642 | 0,6733 | 22 |
| 3 | 0,6692 | 0,7111 | 22 |
| 4 | 0,7074 | 0,7365 | 23 |
| 5 | 0,7832 | 0,7339 | 23 |
| 6 | 0,7540 | 0,7348 | 23 |
| 7 | 0,8802 | 0,7349 | 23 |
| 8 | 0,9683 | 0,7393 | 23 |
| 9 | 0,9383 | 0,7571 | 23 |
| 10 | 1,1049 | 0,7328 | 23 |
| 11 | 1,1040 | 0,7462 | 23 |
| 12 | 1,1186 | 0,7436 | 23 |
| 13 | 1,1721 | 0,7483 | 23 |
| 14 | 1,3357 | 0,7328 | 23 |
| 15 | 1,3225 | 0,7499 | 23 |
| 16 | 1,3276 | 0,7564 | 23 |
| 17 | 1,3778 | 0,7494 | 23 |
| 18 | 1,3575 | 0,7493 | 23 |
| 19 | 1,4086 | 0,7409 | 23 |
| 20 | 1,3976 | 0,7427 | 23 |
| | | Total | 456 |
| Modelo elegido para la evaluación | | | |
| Modelo elegido para la evaluación | Beto_ft_epoch9.model | | |
| Resultados de evaluación | | | |
| Sentimiento | Aciertos | Total | Porcentaje |
| P | 124 | 162 | 76,54% |
| N | 181 | 207 | 87,44% |
| NEU | 43 | 89 | 48,31% |
| Total | 348 | 458 | 75,98% |

Anexo G: Reporte de entrenamiento de modelo para artículos periodísticos en español

| Reporte de Modelo en Español para Artículos | | | | |
|---|--|----------|-------------------------------|---------------------|
| Información del modelo | | | | |
| Nombre del modelo | bert-base-spanish-wwm-uncased | | | |
| Idioma | Español | | | |
| Tipo | Uncased | | | |
| Dataset usado | | | | |
| Nombre | Consolidado de artículos clasificados por la empresa | | | |
| Contenido | Artículos | | | |
| Idioma | Español | | | |
| Longitud | 3394 | | | |
| Sentimientos expresados | Positivo, Negativo, Neutro | | | |
| Comentarios | Se eliminaron los caracteres no alfanuméricos, se mantuvieron las tildes | | | |
| Resultados de entrenamiento | | | | |
| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) | |
| 1 | 0,7446 | 0,6765 | 220 | 3 min 40seg |
| 2 | 0,6639 | 0,7177 | 221 | |
| 3 | 0,6428 | 0,7512 | 221 | |
| 4 | 0,7373 | 0,7269 | 222 | |
| 5 | 0,8152 | 0,7425 | 221 | |
| 6 | 0,9083 | 0,7449 | 221 | |
| 7 | 1,0300 | 0,7441 | 221 | |
| 8 | 1,1497 | 0,7375 | 221 | |
| 9 | 1,2353 | 0,7347 | 221 | |
| 10 | 1,4261 | 0,7204 | 221 | |
| 11 | 1,4016 | 0,7401 | 221 | |
| 12 | 1,3465 | 0,7527 | 221 | |
| 13 | 1,4981 | 0,7399 | 221 | |
| 14 | 1,481 | 0,7528 | 221 | |
| 15 | 1,5984 | 0,7343 | 221 | |
| 16 | 1,5618 | 0,75 | 221 | |
| 17 | 1,5309 | 0,7585 | 221 | |
| 18 | 1,5827 | 0,7553 | 221 | |
| 19 | 1,5947 | 0,7575 | 221 | |
| 20 | 1,6146 | 0,7461 | 221 | |
| | | Total | 4420 | 1 hora 13 min 40seg |
| Modelo elegido para la evaluación | Beto_News_Balanced_1_ft_epoch17.model | | | |
| Resultados de evaluación | | | | |
| Sentimiento | Aciertos | Total | Porcentaje | |
| Positivo | 147 | 169 | 86,98% | |
| Neutro | 125 | 170 | 73,53% | |
| Negativo | 161 | 169 | 95,27% | |
| Total | 433 | 508 | 85,24% | |

Anexo H: Reporte de entrenamiento del modelo optimizado para tweets en español

| Reporte de Modelo Optimizado en Español para Tweets | | | |
|---|--|---|-------------------------------|
| Información del modelo | | | |
| Nombre del modelo | bert-base-spanish-wwm-uncased | | |
| Idioma | Español | | |
| Tipo | Uncased | | |
| Dataset usado | | | |
| Nombre | TASS2019_country_XX_train.xml" | *Se juntaron los datasets de CR, ES,PE y UY | |
| Contenido | Tweets | | |
| Idioma | Español | | |
| Longitud | 3811 | | |
| Sentimientos expresados | P (Positivo), N (Negativo), NEU (Neutral) | | |
| Comentarios | Se eliminaron los caracteres no alfanuméricos, se mantuvieron las tildes | | |
| Resultados de entrenamiento | | | |
| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) |
| 1 | 0,7394 | 0,6274 | 14 |
| 2 | 0,6428 | 0,6552 | 14 |
| 3 | 0,6647 | 0,7274 | 14 |
| 4 | 0,6736 | 0,7394 | 14 |
| 5 | 0,9123 | 0,6796 | 14 |
| 6 | 0,8004 | 0,7655 | 14 |
| 7 | 0,9082 | 0,7580 | 14 |
| 8 | 1,0194 | 0,7488 | 14 |
| 9 | 1,0478 | 0,7684 | 14 |
| 10 | 1,1424 | 0,7575 | 14 |
| 11 | 1,2691 | 0,7332 | 14 |
| 12 | 1,2371 | 0,7532 | 14 |
| 13 | 1,2872 | 0,7366 | 14 |
| 14 | 1,3581 | 0,7441 | 14 |
| 15 | 1,3012 | 0,7536 | 14 |
| 16 | 1,4072 | 0,7549 | 14 |
| 17 | 1,3074 | 0,7692 | 14 |
| 18 | 1,3407 | 0,7595 | 14 |
| 19 | 1,3856 | 0,7553 | 14 |
| 20 | 1,3377 | 0,7592 | 14 |
| | | Total | 280 |
| Modelo elegido para la evaluación | | | |
| Modelo elegido para la evaluación | Beto_2_Adam_ft_epoch17.model | | |
| Resultados de evaluación | | | |
| Sentimiento | Aciertos | Total | Porcentaje |
| P | 123 | 162 | 75,93% |
| N | 173 | 207 | 83,57% |
| NEU | 54 | 89 | 60,67% |
| Total | 350 | 458 | 76,42% |

Anexo I: Reporte de entrenamiento del modelo optimizado para artículos periodísticos en español

| Reporte de Modelo Optimizado en Español para Artículos | | | | |
|--|--|----------|-------------------------------|---------------------|
| Información del modelo | | | | |
| Nombre del modelo | bert-base-spanish-wwm-uncased | | | |
| Idioma | Español | | | |
| Tipo | Uncased | | | |
| Dataset usado | | | | |
| Nombre | Consolidado de artículos clasificados por la empresa | | | |
| Contenido | Artículos | | | |
| Idioma | Español | | | |
| Longitud | 3384 | | | |
| Sentimientos expresados | Positivo, Negativo, Neutro | | | |
| Comentarios | Se eliminaron los caracteres no alfanuméricos, se mantuvieron las tildes | | | |
| Resultados de entrenamiento | | | | |
| Nº Epoca | Validation Loss | F1_Score | Tiempo de entrenamiento (seg) | |
| 1 | 0,5632 | 0,7875 | 230 | 3 min 50seg |
| 2 | 0,5615 | 0,7804 | 230 | |
| 3 | 0,6399 | 0,7885 | 230 | |
| 4 | 0,7698 | 0,7834 | 230 | |
| 5 | 0,8656 | 0,8155 | 230 | |
| 6 | 1,3877 | 0,7560 | 230 | |
| 7 | 1,0742 | 0,8177 | 230 | |
| 8 | 1,2907 | 0,8780 | 229 | |
| 9 | 1,1965 | 0,8099 | 229 | |
| 10 | 1,2899 | 0,8279 | 229 | |
| 11 | 1,1851 | 0,8209 | 229 | |
| 12 | 1,2150 | 0,8188 | 229 | |
| 13 | 1,3017 | 0,8110 | 229 | |
| 14 | 1,2709 | 0,8215 | 229 | |
| 15 | 1,3107 | 0,8148 | 229 | |
| 16 | 1,3056 | 0,8213 | 229 | |
| 17 | 1,3135 | 0,8172 | 229 | |
| 18 | 1,3145 | 0,8174 | 229 | |
| 19 | 1,3082 | 0,8174 | 227 | |
| 20 | 1,3092 | 0,8174 | 226 | |
| | | Total | 4582 | 1 hora 16 min 22seg |
| Modelo elegido para la evaluación | Beto_News_Ada_3_t_epoch8.model | | | |
| Resultados de evaluación | | | | |
| Sentimiento | Aciertos | Total | Porcentaje | |
| Positivo | 146 | 169 | 86,39% | |
| Neutro | 141 | 170 | 82,94% | |
| Negativo | 162 | 169 | 83,43% | |
| Total | 449 | 508 | 88,39% | |

Anexo J: Repositorio del código de la aplicación para pruebas de usuario

Los enlaces donde se puede encontrar el repositorio del código necesario para clonar la aplicación desarrollada son:

- *Frontend*: <https://gitlab.com/gabozarate14/proyecto-de-tesis>
- *Backend*: <https://gitlab.com/gabozarate14/proyecto-de-tesis-back>

Anexo K: Bases de datos textuales

El anexo es la ruta de una carpeta pública de Google Drive, en la cual se encuentran las bases de datos textuales generadas y posteriormente clasificadas para poder cumplir el tercer objetivo de la presente tesis. El enlace es el siguiente:

https://drive.google.com/drive/folders/1U-Et3Bg5bG8AUu7d0uaHq_BdDhQ5iBu4?usp=sharing