INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Laboratorio de Ciencias Cognitivas Computacionales

Impacto del *Leet Speaking* en la detección de tendencias suicidas en *tweets*

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRÍA EN CIENCIAS DE LA DE COMPUTACIÓN

P R E S E N T A:

Ing. Cristian Camilo Segura Morales

Director de tesis:

Dr. Francisco Hiram Calvo Castro

México, CDMX Enero 2023

SIP-13
REP 2017

INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS
Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México, a 29 de 11 del 2022

El Colegio de Profesores de Posgrado del **Centro de Investigación en Computación** en su Sesión
(Unidad Académica)

Extraordinaria No 16 celebrada el día 25 del mes octubre de 2022, conoció la solicitud presentada por el (la) alumno (a):

Apellido Paterno:	SEGURA	Apellido Materno:	MORALES	Nombre (s):	CRISTIAN CAMILO

Número de registro: A 2 1 0 3 6 6

del Programa Académico de Posgrado: **Maestría en Ciencias de la Computación**

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

"Impacto del Leet Speaking en la detección de tendencias suicidas en tweets"

Objetivo general del trabajo de tesis:

Comprobar cómo la modificación de caracteres en palabras (leet speaking) afecta el reconocimiento de tuits con tendencias suicidas. Adicionalmente, implementar un componente que transforme este tweet codificado en texto claro para su correcta clasificación.

2.- Se designa como Directores de Tesis a los profesores:

Director: **Dr. Francisco Hiram Calvo Castro** 2° Director:

No aplica: X

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

Centro de Investigación en Computación

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director(a) de Tesis

Dr. Francisco Hiram Calvo Castro

Aspirante

C. Cristian Camilo Segura Morales

2° Director de Tesis

Presidente del Colegio

Dr. Francisco Hiram Calvo Castro

IPN-CIC

SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 16:00 horas del día 14 del mes de diciembre del 2022 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: Centro de Investigación en Computación para examinar la tesis titulada:

"Impacto del Leet Speaking en la detección de tendencias suicidas en tweets" del (la) alumno (a):

Apellido Paterno:	SEGURA	Apellido Materno:	MORALES	Nombre (s):	CRISTIAN CAMILO

Número de registro: A 2 1 0 3 6 6

Aspirante del Programa Académico de Posgrado: **Maestría en Ciencias de la Computación**

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 04 % de similitud. **Se adjunta reporte de software utilizado**.

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** ☐ **NO** ☒ **SE CONSTITUYE UN POSIBLE PLAGIO.**

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*
Coincidencias menores relacionadas con la bibliografía.

**Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.

Finalmente y posterior a la lectura, revisión individu al, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** ☒ **SUSPENDER** ☐ **NO APROBAR** ☐ la tesis por **UNANIMIDAD** ☒ o **MAYORÍA** ☐ en virtud de los motivos siguientes:
El trabajo es adecuado para obtener grado de maestría.

COMISIÓN REVISORA DE TESIS

Dr. Francisco Hiram Calvo Castro
Director de Tesis

Dr. Alexander Gelbukh

M. en C. José Eduardo Valdez Rodríguez

Dra. Olga Kolesnikova

Dr. Grigori Sidorov

Dr. Miguel Jesús Torres Ruiz

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
DIRECCIÓN

Dr. Francisco Hiram Calvo Castro
PRESIDENTE DEL COLEGIO DE PROFESORES

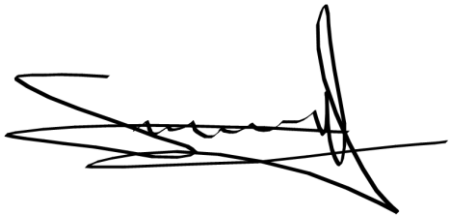
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 4 del mes de enero del año 2023, el (la) que suscribe Cristian Camilo Segura Morales alumno(a) del programa Maestría en Ciencias de la Computación con número de registro A210366, adscrito(a) al Centro de Investigación en Computación manifiesta que es autor(a) intelectual del presente trabajo de tesis bajo la dirección del Dr. Francisco Hiram Calvo Castro y cede los derechos del trabajo titulado Impacto del Leet Speaking en la detección de tendencias suicidas en tweets, al Instituto Politécnico Nacional, para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expresado del autor y/o director(es). Este puede ser obtenido escribiendo a las siguiente(s) dirección(es) de correo. cseguram2021@cic.ipn.mx, cricasemo@gmail.com. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.



Cristian Camilo Segura Morales

Resumen

El Leet Speaking se ha impuesto como una técnica sencilla de ofuscación de información principalmente en jóvenes y adolescentes en las redes sociales. Su objetivo: camuflar el sentido del mensaje a transmitir. Se lleva a cabo mediante la modificación de algunos caracteres en el cuerpo del texto con otros que tengan características visuales similares al carácter que se desea reemplazar, para que al momento de realizar la lectura, no modifique el sentido del mensaje a comunicar. Esta codificación no es tenida en cuenta como una variable en las tareas de procesamiento de lenguaje natural actuales, pero sí impactan directamente en el desempeño de los sistemas.

En el presente trabajo se demostró el impacto real de esta técnica en textos publicados en la res social *twitter* (tweets) que pudieran contener información indicativa de tendencias suicidas, utilizando específicamente el modelo pre-entrenado BERT como herramienta tecnológica en la predicción. El problema fue planteado con la siguiente premisa: entre más palabras o caracteres involucre la técnica de ofuscación, el desempeño del clasificador se verá proporcionalmente afectado de manera negativa. Utilizando el clasificador indicado y realizando la modificación sobre datos que el modelo no haya visto en la fase de entrenamiento, se realizaron ajustes en el contenido de los tuits de acuerdo con las fases experimentales definidas. Posteriormente, se presenta el diseño y el desarrollo de un componente que mediante el uso de modelos de lenguaje, que permitió realizar la transformación de los tuits en formato *leet speaking* a texto en claro, para obtener de este modo los datos sin ningún tipo de ofuscamiento. El resultado obtenido fue ingresado nuevamente al modelo de clasificación, para comprobar que los tuits fueron catalogados de manera correcta conforme a la etiqueta de tendencia asignada a cada uno de ellos.

Como ultima parte del trabajo, se realizó el análisis de los datos obtenidos experimentalmente, concluyendo que, el modelo implementado es una solución viable al problema planteado. Finalmente, se realiza la propuesta de trabajo futuro con ciertas variables que no fueron contempladas en esta investigación, pero que permiten robustecer la solución y abarcar otro tipo de modificaciones en el contenido compartido por los usuarios de *twitter* y de otras redes sociales no incluidas.

Abstract

Leet Speaking has been imposed as a simple information obfuscation technique, mainly among young people and adolescents on social networks. Its objective: camouflaging the meaning of the message to be transmitted. It is carried out through modifying some characters in the body of the text with others that have similar visual characteristics to the character being replaced. So, when the text is readed, it does not modify the meaning of the message to be communicated. This encoding is not taken into account as a variable in current natural language processing tasks, but it does have a direct impact on system performance.

In this work, the real impact of this technique on the prediction of suicidal tendencies in tweets was demonstrated using specifically the BERT pre-trained model as a technological tool in the prediction. The problem is defined with the premise that if more words or characters are involved in the obfuscation technique, the performance of the classifier is negatively affected. Using the indicated classifier and making the modification on data that the model has not seen in the training phase, adjustments were made to the content of the tweets according to the defined experimental phases. Subsequently, the design and development of a component is presented that through the use of language models, which allowed the transformation of the tweets in leet format to plain text to obtain the data without any type of daze. The result obtained was entered into the classification model again, to verify that the tweets were classified correctly according to the trend label assigned to each of them.

At last, the analysis of the data obtained experimentally was carried out, with which it is possible to conclude that the implemented model is a valid solution to the problem posed. Finally, the proposal for future work is made with certain variables that were not considered in this implementation, but that allow the solution to be strengthened and cover other types of modifications in the content shared by users of twitter and other social networks not contemplated.

Agradecimientos

Agradezco el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACYT, México), el Instituto Politécnico Nacional y el Centro de Investigación en Computación, estas instituciones hicieron posible el desarrollo de esta tesis.

A mi familia quien si su apoyo y motivación no habría podido conseguir este nuevo título en mi vida académica y profesional.

A mis amigos del CIC, Maraya, Beto, Emilio, Cesar, Miguel, Arturo, Tania, Dianita quienes fueron de gran motivación y que con su ayuda, apoyo, palabras de aliento y respaldo, hacen parte también de este triunfo.

A todos ustedes, gracias.
Cristian Camilo Segura Morales

Índice general

Índice de figuras

Índice de tablas

Lista de código fuente

Parte I

Introducción

Capítulo 1

Introducción

1.1 Antecedentes

El uso de las redes sociales ha crecido de una manera imponente en los últimos años [3]. Las redes sociales son definidas como los sitios y las herramientas en línea que facilitan la interacción entre los diferentes usuarios permitiéndoles la oportunidad de intercambiar información, opiniones, intereses e incluso emociones. Dentro de los principales usos de las redes sociales se encuentran el entretenimiento, comunicación, y el creciente flujo de datos desde y hacia los usuarios, entre otros. Con el auge de estas plataformas sociales, una gran cantidad de horas diarias son invertidas de manera virtual en herramientas como Instagram, Facebook, twitter, YouTube, WhatsApp entre muchas otras. Algunos autores sugieren que las redes sociales han cambiado la forma de la interacción entre individuos y el comportamiento de los usuarios de manera individual y colectiva alrededor del globo [4, 5].

Existen diferentes investigaciones que indican que cuando se experimenta una fuerte cantidad de sentimientos positivos o negativos, hay una gran motivación de querer compartirlos con los demás [6]. De acuerdo con la teoría del intercambio social de emociones de Rimé, compartir las emociones ayuda en dos objetivos: la satisfacción de las necesidades socioemocionales, que consiste en obtener apoyo de terceros para revalidar los sentimientos, normalizar las experiencias y sentir estados emocionales positivos. Y como segundo objetivo, se encuentran las necesidades cognitivas, que implica recibir consejos de los demás para ayudar a dar sentido a las experiencias vividas. Las redes sociales son uno de los principales motores para ese intercambio, que además el auge de las mismas ha contribuido

enormemente en la globalization y la ruptura de fronteras, de modo que todo lo que compartes puede hacerse global en cuestión de horas, muestra clara de ello es la aparición de los influencers, las temas que se hacen tendencia y los contenidos virales

Como lo indica [7], el suicidio es la segunda mayor causa de muerte entre adolescentes hombres entre los 15 a 29 años, y la primera causa de muerte en mujeres jóvenes de 15 a 19 años. Esto debido principalmente a que las personas jóvenes son las más susceptibles a adoptar comportamientos suicidas que puedan inducir otras personas; sobre aquellas que ya posean previamente este tipo de tendencias [8]. El uso prolongado de las redes sociales conlleva al desarrollo de características sicológicas negativas obteniendo como consecuencia a un incremento de problemas de salud mental incluyendo la depresión [9], que posteriormente puede desencadenar patrones de ansiedad, desespero y finalmente las tendencias suicidas de los individuos [10]. Otra característica generada por las redes sociales es la impulsividad [11], originada precisamente por la interacciones reiterativas en internet y algunos de los retos que allí se presentan, que incitan en los usuarios ciertos comportamientos sociales delicados como el *cyberbullying* [12], y que influyen de manera negativa aumentando la manisfectación de las tendencias y prácticas suicidas en jóvenes.

Debido a esta creciente predisposición de compartir emociones a través de conversaciones publicas y/o privadas en las redes sociales y el aumento en las técnicas utilizadas para la detección de publicaciones potencialmente peligrosas en ellas, las empresas de tecnología implementan herramientas moderadoras con el objetivo de evitar que este tipo de contenido sea compartido fácilmente [13]. De allí, surge que los usuarios opten por estrategias para evadir estos filtros de moderación y compartir sus publicaciones sin algún tipo de restricción. Como consecuencia, surge la estrategia del *leet speaking*, que consiste en la sustitución de letras por símbolos y/o números que lucen como la letra a ser reemplazada. Este cambio no genera alteración sobre la intención del mensaje a transmitir. Esta técnica no requiere de experiencia del usuario en herramientas tecnológicas para su entendimiento ni ningún tipo de conocimiento técnico avanzado. Inclusive, para aquellas personas que jamás han escuchado o no saben en qué existe un nombre formal para técnica, puede ser entendido e implementado con facilidad ya que está asociado a una característica de comunicación básica que es la escritura de texto.

1.2 Planteamiento del problema

Los procesadores de lenguaje utilizados para filtrar el contenido de las publicaciones en *twitter* no reconocen publicaciones realizadas con técnicas de *leet speaking*, lo que permite que mensajes escritos con esta técnica, que incumplen las políticas de privacidad y protección al usuario permanezcan en la red social.

1.3 Hipótesis

La implementación de un componente que utiliza técnicas de procesamiento de lenguaje natural, permite realizar la conversión de tuits publicados con técnicas de *leet speaking* a texto en claro, para ejecutar una correcta detección de aquellas publicaciones que en su contenido incumplas los parámetros establecidos por las compañías, como los filtros lenguaje abusivo, racismo, que indiquen intenciones suicidas entre otros.

1.4 Objetivos

1.4.1 Objetivo general

Transformar tuits escritos con técnicas de *leet speaking* para detectar si estas publicaciones compartidas por los usuarios indican intenciones suicidas (filtro elegido para la realización de este estudio), a partir de la afirmación de que la modificación intencional de su estilo de lenguaje y/o contenido afecta de manera negativa el desempeño de los modelos de inteligencia artificial actuales utilizados para la detección de estas tendencias.

1.4.2 Objetivos específicos

- Entrenar un transformador tipo BERT que clasifique de manera precisa lenguaje que pueda sugerir tendencias suicidas en tuits.
- Demostrar que las técnicas de *leet speaking* en tuits afectan el desempeño de clasificación en el transformador entrenado.
- Diseñar un prototipo para la transformación de tuits escritos con técnicas de *leet speaking* a texto en claro.
- Desarrollar el prototipo diseñado con las herramientas tecnológicas adecuadas.

- Probar el componente implementado con los tuits escritos con técnicas de *leet speaking*.

1.5 Alcances del trabajo

El alcance del presente trabajo se limitó a realizar modificaciones *leet* en publicaciones realizadas por diferentes usuarios en la red social *twitter*. El cambio en el contenido se realizó mediante la sustitución de un único carácter, por su correspondiente transformación a un número o un símbolo. Es decir, no hay asociación de más de un carácter a una letra. Esta modificación fue efectuada exclusivamente sobre los tuits clasificados con tendencias suicidas de acuerdo con el conjunto de datos seleccionado para el análisis. Aquellos tuits que no incluían contenido asociado a tendencias suicidas, no sufrieron de ningún tipo de modificación en sus enunciados.

Como modelos de lenguaje para las tareas de clasificación y conversión, se realizó uso de modelos preentrenados debido a sus buenos desempeños en las tareas de procesamiento de lenguaje natural. Para entrenamiento y prueba del modelo de clasificación, se seleccionó el modelo BERT; y para la fase de conversión de contenido *leet*, se utilizaron tres variantes del modelo GPT2 (GPT2 estándar, GPT2 destilado, y GPT2 mediano).

La fase de experimentación se realizó con tres conjuntos de datos que son los tuits originales, los tuits con las modificaciones *leet speaking* y los tuits resultantes del proceso de transformación. Con estos resultados y mediante la generación de métodos de comparación como gráficas y matrices de confusión, fue posible determinar si la solución desarrollada satisfizo los objetivos de investigación cumpliendo con la hipótesis planteada.

1.6 Justificación

Debido a que la gran mayoría de herramientas actuales en la detección de contenidos potencialmente peligroso, no contemplan las técnicas de modificación de contenido mediante el modelo de *leet speaking*, muchos de los usuarios pueden compartir contenido que no se encuentra dentro del lineamiento establecidos por *twitter* al momento de realizar las diferentes publicaciones en esta red social.

Al construir un prototipo articulado con herramientas de modelos de lenguaje que permitan transformar este tipo de contenido, es posible incrementar la precisión con la que

se detectan las infracciones a la política de protección al usuario de dicha red social [1]. De esta manera, identificar este grupo de usuarios de manera temprana, permitiría realizar un seguimiento oportuno a casos particulares para así aplicar los controles establecidos hacia los contenidos marcados como peligrosos.

Los trabajos realizados en favor de la salud mental, usualmente pueden no llegar a generar mucho interés por las áreas comerciales de las organizaciones debido a que no se encuentran alineados con sus estrategias de negocios. Es por ello, que los pocos investigadores que se atreven a entrar en estas líneas del trabajo, cuentan con recursos limitados para desempeñar su tarea. De esta manera, el prototipo propuesto contempla ser un elemento de apoyo eficiente, que evita incurrir en gastos adicionales por la aparición de una variable omitida en el análisis inicial de los diferentes modelos, dando como valor agregado de la solución, apoyo a los investigadores en su trabajo y la versatilidad de la herramienta que puede aplicarse para identificar otros tipos de lenguajes que surjan en el ejercicio de las redes sociales, como el uso de abreviaturas por ejemplo.

1.7 Organización del escrito

Esta tesis se divide en seis partes:

- En la Parte 1. Introducción, se presenta el objetivo del trabajo, alcances y limitaciones. Se ofrece un breve acercamiento al impacto de las redes sociales en los comportamientos de los usuarios y de cómo estos utilizan técnicas para llegar a compartir contenidos sin ninguna restricción.

- En la Parte 2. Antecedentes y marco teórico, Se abarcan las bases teóricas en las que se sustenta este trabajo, dando un recorrido por aquellos aspectos cognitivos, y aquellos técnicos y tecnológicos utilizados para las etapas de comprobación de la hipótesis, desarrollo, implementación y pruebas del prototipo propuesto. Adicionalmente, se da un recorrido sobre el estado del arte, donde se exponen trabajos de investigación asociados a la identificación de tendencias suicidas en diferentes redes sociales. Además, métricas de evaluación y desempeño de estos trabajos e investigaciones.

- En la Parte 3. Desarrollo, se presenta el detalle del trabajo realizado en esta tesis.

[1] https://twitter.com/es/privacy

Iniciando con la explicación para la comprobación de la hipótesis, la división del problema posterior a la comprobación, el diseño de las diferentes etapas que componen el prototipo de solución detallando cada una de sus partes, y se explica cómo funcionan e interactúan cada una de las piezas con las demás. Finalmente, el método de evaluación para verificar que la solución desarrollada se comporta de acuerdo con lo planteado.

- En la Parte 4. Resultados experimentales y discusión, se muestra el diseño, aplicación y análisis de resultados de las pruebas para la etapa de comprobación de la hipótesis, y las pruebas dirigidas a verificar el correcto funcionamiento del prototipo desarrollado.

- En la Parte 5. Conclusiones y trabajo futuro, comprende las aseveraciones finales obtenidas a partir del análisis de los resultados experimentales y la verificación del cumplimiento los objetivos propuestos. Además, se plantean opciones de trabajo a futuro surgidas a partir de las variables derivadas de la implementación desarrollada.

- Finalmente, se presentan las referencias bibliográficas que fueron utilizadas como apoyo en los procesos de análisis e implementación del trabajo.

Parte II

Antecedentes y marco teórico

Capítulo 2

Marco teórico

2.1 *Leet speaking*

Los orígenes de esta técnica se encuentras ligados a las actividades de *hacking* y *cracking* alrededor de los años 80. Es posible identificar en la red diferentes tipos de usuarios de acuerdo a las necesidades personales de cada uno, y a aquellos que se "sumergen" en instancias más profundas, suelen ser considerados usuarios "élite", debido al tipo de información que puede encontrarse y los riesgos inherentes al hacer este tipo de inmersión.

Estos sitios suelen ser altamente vigilados por entidades públicas y privadas para evitar la proliferación y masificación de actividades ilegales en la red, es por ello que estos usuarios "élite" tenían que buscar formas de ocultar su existencia en la red y el rastreo de las actividades que realizaban. De esta manera, surge el *leet speaking*, considerado como un lenguaje que se encuentra en un evolutivo y contante cambio, con el objetivo de ofuscar de manera intencional registros textuales para evadir la detección y seguimiento de la intención real a transmitir.

La palabra *leet* proviene proviene de la derivación del fonema escrito con errores ortográficos de la palabra *elite (Eleet)*. Con este sistema, se realiza la transformación de los grafemas normales en un registro textual, por otros que sean similares visualmente y que no cambien la intención del mensaje a transmitir. Este sistema cuenta con diferentes versiones, siendo la más sencilla de ellas, un reemplazo individual de ciertos caracteres por números; llegando a un mayor nivel de complejidad, donde un símbolo puede ser sustituido por dos o más caracteres ASCII en su reemplazo. En la figura 2.1 se presenta un ejemplo de una transformación sencilla y una transformación compleja para una misma frase.

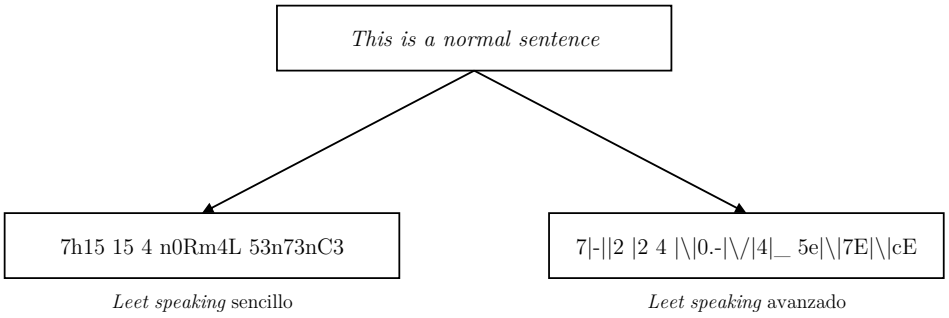


Figura 2.1: Ejemplos de transformaciones *leet speaking*

2.2 Modelos para representar emociones

Los modelos de emociones son fundamentales para los sistemas detectores de las mismas ya que definen cómo pueden ser representadas. Estos modelos afirman que las emociones pueden presentarse en varios estados por lo cual nace la necesidad de distinguir cada uno de ellos; de modo que, cuando se genera cualquier tipo de actividad concerniente a la detección de emociones es importante realizar la definición del modelo a utilizar. Existen diferentes modelos que representan emociones, pero se hará énfasis en los modelos de emociones discretos y dimensionales.

2.2.1 Modelos de emociones discretos (*Discrete emotion Models - DEMs*)

Clasifican las emociones en diferentes categorías, donde se toman los siguientes tres modelos a consideración:

Modelo de Paul Ekman [14]. Realiza diferenciación de emociones con un conjunto de seis categorías básicas. En su modelo afirma que existen seis emociones originadas desde diferentes sistemas neuronales como el resultado de cómo se percibe una situación, por ello son independientes. Estas emociones son ira, felicidad, sorpresa, tristeza, temor y desagrado. Adicionalmente, estas emociones en conjunto pueden generar otras más complejas como la vergüenza, orgullo, lujuria, culpa, avaricia, deseo, celos entre otras.

Modelo de Plutchik y Kellerman [15]. Afirma también que existen emociones primarias las cuales ocurren en pares opuestos y se encargan de producir emociones más complejas con sus combinaciones. Nombra un total de ocho emociones primarias en pares fundamentales las cuales son alegría frente a tristeza, confianza frente a disgusto, ira frente

a miedo y sorpresa frente a anticipación. De acuerdo con Plutchik, para cada una de las emociones existen varios grados de intensidad que ocurren de acuerdo a como los eventos son construidos por un individuo.

Modelo de Orthony, Clore y Collins (*OCC Model*) [16]. Afirman que las emociones son el resultado de cómo un individuo percibe los eventos y esas emociones varían de acuerdo con el grado de intensidad en el que son percibidas. Realizan la diferenciación de 22 emociones definiendo su modelo de la siguiente manera: ira, felicidad, sorpresa, tristeza, temor, desagrado, envidia, desagrado, agrado, dolor, alivio, envidia, reproche, autorreproche, aprecio, vergüenza, lástima, admiración, decepción, esperanza, miedos reafirmados, agradecimiento.

2.2.2 Modelos de emociones dimensionales (*Dimensional emotion models DiEMs*)

El modelo dimensional supone que las emociones no son independientes unas de las otras y que existe una relación entre ellas, por lo que surge la necesidad de posicionarlas de manera conjunta en un espacio dimensional que muestre qué tan relacionadas están las emociones y realizando una representación de los dos estados conductuales básicos fundamentales: bien y mal. Los siguientes son los dos modelos básicos dimensionales que se utilizarán:

El modelo de Rusell [17] consiste en una estructura de dos dimensiones denominado circunferencia de afecto. El modelo define dos valores de medida, *valence* que diferencia las emociones en un rango de positiva a negativa, y *arousal* las diferencia en un contexto de emoción o apatía. Esta relación se muestra en la figura 2.2 sección izquierda.

El modelo de Plutchik presenta las emociones en un disco bidimensional con *valence* en el eje vertical y *arousal* en el horizontal. La figura 2.2 en la sección derecha, muestra este esquema de representación en capas, donde las más internas corresponden a derivaciones de las ocho emociones primarias, seguidas precisamente por estas emociones primarias y en la parte exterior combinaciones de las emociones primarias. Según la posición en la figura, se determina qué tan relacionadas se encuentran las emociones.

En consecuencia al problema a resolver se selecciona el modelo emocional a trabajar. Los modelos de emoción discretos usualmente son los seleccionados para los problemas de clasificación de emociones debido a la simplicidad, pero carecen de una gama más amplia de clase de emociones, la intensidad y grado de ocurrencia en contraste con los modelos dimensionales. Pero los modelos dimensionales son recomendables cuando se

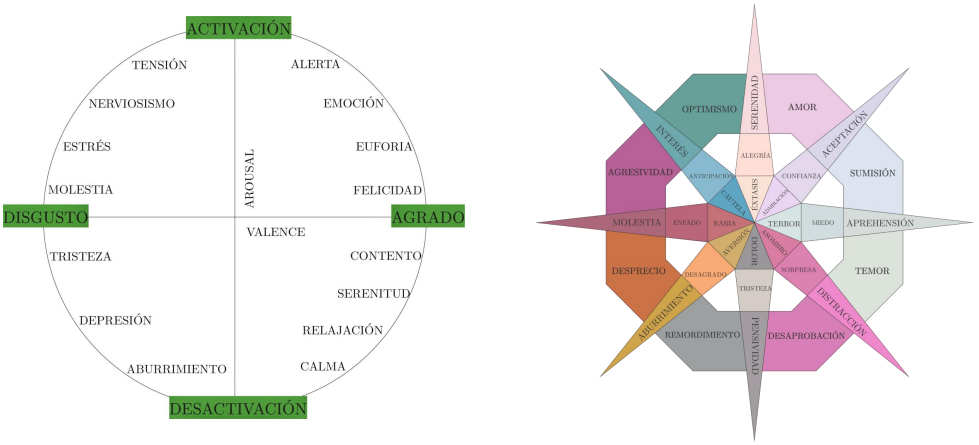


Figura 2.2: Modelos de emociones de Rusell (izquierda) y Plutchik (derecha)

Tabla 2.1: Emociones básicas usadas en la clasificación

Sigla	Clase de emoción
A	Enojo
D	Disgusto
F	Temor
H	Felicidad
Sa	Tristeza
Su+	Sorpresa positiva
Su-	Sorpresa negativa

desea categorizar una mayor cantidad de emociones y se quiere diferenciar entre emociones que son muy similares entre sí.

2.3 Recursos utilizados en el reconocimiento de emociones

Comúnmente en el reconocimiento de emociones se utilizan dos tipos de conjunto de datos, los *corpora* que corresponden al conjunto de textos completos clasificados según la emoción a la que se encuentra asociado cada uno de ellos, y los *lexicons* que corresponden a un conjunto de listas ordenadas con palabras asociadas cada una de ellas a una emoción.

2.3.1 Corpora

[18][1] Se presentan un total de 185 historias infantiles donde la emoción se clasifica a nivel de las oraciones que se encuentran en cada una de las historias y se asigna una etiqueta a cada una de las frases de acuerdo con el cuadro 2.1.

[1] people.rc.rit.edu/~coagla/affectdata/index.html

Para la clasificación se realiza la asignación en pares de las mismas historias y con el objetivo de evitar sesgos cada uno de los evaluadores trabaja de manera separada y realiza la asignación de una sigla de manera independiente. Cuando existe un desacuerdo en la etiqueta asignada, el evaluador principal elige una de las etiquetas contendientes.

El corpus de [19] cuenta con un conjunto de publicaciones en diferentes blogs que son obtenidas mediante el uso de palabras semilla para asignar a cada emoción. Por ejemplo, palabras como miedo, susto, o pánico son clasificados en la categoría de temor. La caracterización se realiza a nivel de oración en una de las ocho categorías clasificadoras (felicidad, tristeza, enojo, desagrado, sorpresa, temor, emociones combinadas y sin emoción).

[20][2] describen el resultado de la aplicación de una encuesta internacional de antecedentes de emociones y reacciones (Internations Survey on Emotion Antecedents and Reactions – ISEAR); un selecto grupo de psicólogos de todo el mundo trabajó en este proyecto y adicionalmente, hubo una participación aproximada de 3000 estudiantes indicando situaciones del día a día en las que han experimentado emociones de alegría, miedo, temor, tristeza, desagrado, vergüenza y culpa.

En SemEval-2007 [21][3] se cuenta con titulares de noticias obtenidos de diferentes ejemplares, como BBC, CNN, New York Times y el motor de búsqueda de noticias de Google. La estructura con la que cuenta un titular de noticias permite realizar la clasificación a nivel de oraciones. Cada una es clasificada de acuerdo con una o más de las siguientes emociones: enojo, disgusto, temor, alegría, tristeza y sorpresa. La implementación cuenta con dos conjuntos de datos uno para entrenamiento con 250 titulares, y uno de prueba con un total de 1000 titulares.

SemEval-2018 [22][4] incluye la recopilación de tuits, donde se clasifican de tipo neutral o si expresan una o más emociones dentro de las que se encuentran enojo, desagrado, temor, alegría, amor, optimismo, pesimismo, tristeza, sorpresa y confianza. De igual manera existe diferenciación entre los conjuntos de entrenamiento. Los tuits fueron obtenidos en tres idiomas: inglés, árabe y español.

SemEval-2019 [23][5] presenta diálogos textuales entre dos personas donde el primero de ellos es quien inicia las conversaciones, luego el segundo responde y de nuevo el primero vuelve conversar en un esquema de turnos. Cada una de las conversaciones es etiquetada

[2] github.com/PoorvaRane/Emotion-Detector/blob/master/ISEAR.csv
[3] web.eecs.umich.edu/~mihalcea/affectivetext/#resources
[4] competitions.codalab.org/competitions/17751
[5] www.humanizing-ai.com/emocontext.html

de acuerdo con la emoción que expresa; alegría, enojo, tristeza entre otras. La clasificación de la emoción es realizada con base en la interacción en el tercer turno de la conversación.

SemEval-2020 [24][6] aborda la clasificación de tuits en tres grupos principales de emociones: positivos que expresan felicidad, exaltar a una persona, grupo, país o celebrar algo. Negativos que expresan ataques a alguna persona, grupo, producto, o país, también disgusto, o tristeza hacia algo y crítica. Y neutrales que corresponden a noticias o publicidad.

En el cuadro 2.2 se realiza la clasificación de los conjuntos de datos especificados indicando el número de emociones según el tipo de emoción detectado y el número de ocurrencias de cada una de las emociones encontradas.

2.3.2 Lexicones

Existen diversos lexicones, también conocidos como diccionarios, que son utilizados comúnmente para el análisis de emociones. A continuación se describen los recursos existentes.

WordNet[7] se trata de una base de datos léxica en inglés en línea. Agrupa verbos, sustantivos, adjetivos, y adverbios en conjuntos de sinónimos denominados *synsets*.

WordNet-Affect[8] es una extensión de *WordNet*; un subconjunto de los *synsets* de éste con palabras que expresan emociones directas o indirectas, se utilizan etiquetas semánticas para su asignación.

SentiWordNet[9] asigna una de los tres posibles valores de clasificación positivos, negativos, u objetivos a cada uno de los *synsets* de *WordNet*

AFINN[10] consta de palabras calificadas de manera manual para el valor *valence* con un número entero entre menos cinco (negativo) y más cinco (positivo).

NRC Word-Emotion Association Lexicon[11] fue creado de manera manual utilizando el sistema de turcos mecánicos de Amazon. Ocho emociones, que son enojo, anticipación, disgusto, miedo, alegría, tristeza, sorpresa y confianza, y las categorías de sentimientos, positivos y negativos están incluidos.

NRC Affect Intensity Lexicon[12] proporciona puntuaciones de intensidad con valores reales por las emociones enojo, miedo, tristeza y alegría.

[6] paperswithcode.com/paper/semeval-2020-task-9-overview-of-sentiment
[7] wordnet.princeton.edu/download/current-version
[8] multiwordnet.fbk.eu/english/home.php
[9] github.com/aesuli/SentiWordNet
[10] github.com/fnielsen/afinn
[11] saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
[12] www.saifmohammad.com/WebPages/AffectIntensity.htm

Tabla 2.2: Corpora para reconocimiento de emociones en textos.

Emoción	**Alm et al**	**Aman**	**ISEAR**	**SemEval 2007**		**SemEval 2018**		**SemEval 2019**		**SemEval 2020**	
				Train	***Test***	***Train***	***Test***	***Train***	***Test***	***Train***	***Test***
Alegría	445	536	1094	35	113	2477	1442	4243	284	-	-
Amor	-	-	-	-	-	700	516	-	-	-	-
Anticipación	-	-	-	-	-	978	425	-	-	-	-
Confianza	-	-	-	-	-	357	153	-	-	-	-
Culpa	-	-	1093	-	-	-	-	-	-	-	-
Disgusto	-	172	1096	9	12	2602	1099	-	-	-	-
Enojo	-	179	1096	20	21	2544	1101	5506	298	-	-
Enojo-Disgusto	218	-	-	-	-	-	-	-	-	-	-
Miedo	166	115	1095	32	92	1242	485	-	-	-	-
Neutral	-	2800	-	146	650	2207	75	-	-	3974	206
Optimismo	-	-	-	-	-	1984	1143	-	-	6005	3061
Otros	-	-	-	-	-	-	-	14948	4677	-	-
Pesimismo	-	-	-	-	-	795	375	-	-	2023	522
Sorpresa	114	115	-	8	42	361	170	-	-	-	-
Tristeza	264	173	1096	41	104	2008	960	5463	250	-	-
Vergüenza	-	-	1096	-	-	-	-	-	-	-	-
Total	1207	4090	7666	291	1034	18255	7944	30160	5509	12002	3789

NCR Valence, *Arousal* y *Dominance* Lexicon[13] incluye una lista de más de 20.000 palabras y sus valores de *valence*, *arousal* y *dominance*. Las puntuaciones van de 0 a 1.

NRC Hashtag Emotion Lexicon[14] es un lexicón generado de manera automática a partir de tuits que incluyen *etiquetas* de palabras de emoción, como `#happy`. Asocia las palabras con las emociones ira, disgusto, miedo, tristeza, anticipación, sorpresa, alegría y confianza.

NRC Hashtag Sentiment Lexicon[15] fue creado de manera automática a partir de tuits que incluyen etiquetas de palabras de sentimiento como `#amazing`. Asocia palabras con un sentimiento positivo o negativo.

Sentiment140 Lexicon [16] fue generado también automáticamente a partir de tuits con emoticonos.

Finalmente, *Spanish Emotion Lexicon* (SEL)[25] [17] Contiene un total de 2036 palabras las cuales se encuentran asociadas con el factor de probabilidad de uso afectivo (FPA) con respecto a por lo menos una de las siguientes emociones básicas: alegría, ira, miedo, tristeza, sorpresa y disgusto. [26]

2.4 Técnicas utilizadas en el reconocimiento de emociones en textos

Se realiza la distinción entre cuatro enfoques utilizados para reconocer las emociones en el texto que son los basados en palabras clave, basados en reglas, basados en el aprendizaje y basados en aprendizaje profundo. Los artículos se clasifican con base en el enfoque propuesto, el uso de dicha clasificación ayudará a evaluar estos enfoques en función de su desempeño, fortalezas y limitaciones y establecer una comparación entre ellos. Las emociones explícitas son reconocidas principalmente con enfoques basados en palabras clave. Los otros tres se utilizan principalmente para reconocer las emociones implícitas en el texto, a pesar de que también han sido utilizados para el reconocimiento de emociones explícitas.

[13] saifmohammad.com/WebPages/nrc-vad.html
[14] saifmohammad.com/WebPages/lexicons.html#NRCtwitter
[15] saifmohammad.com/WebPages/lexicons.html#NRCtwitter
[16] help.sentiment140.com/for-students
[17] https://www.cic.ipn.mx/~sidorov/

2.4.1 Enfoque basado en palabras clave

Este enfoque realiza la búsqueda de ocurrencias de palabras claves en un texto dado y realiza la asignación de una etiqueta de emoción de acuerdo con la palabra clave que ha sido detectada. La aproximación más utilizada es la técnica denominada *keyword-spotting*.

En la figura 2.3 se presenta el esquema general del enfoque en palabras clave.

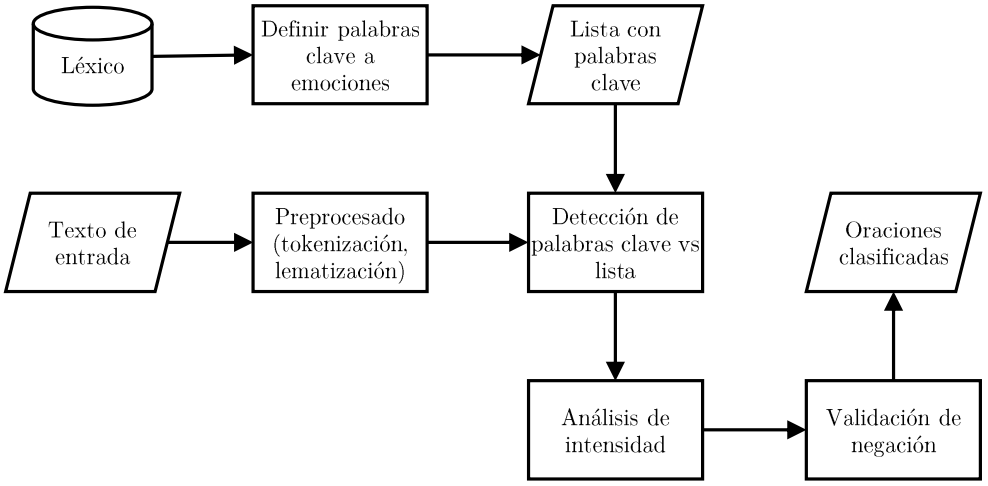


Figura 2.3: Enfoque basado en palabras clave

2.4.2 Enfoque basado en reglas

Se centra en la manipulación de diferentes fuentes de conocimiento para realizar la interpretación de la información de entrada y ejecutar su correspondiente clasificación. En primer lugar, se inicia el preprocesamiento del texto de entrada, donde se aplica tokenización, eliminación de palabras gramaticales, lematización, etiquetado gramatical y análisis de dependencias. Luego, las reglas de emoción son extraídas utilizando conceptos lingüísticos, estadísticos y computacionales. Las mejores reglas son seleccionadas y finalmente son aplicadas al texto de entrada para determinar cada una de las etiquetas de emoción.

En la figura 2.3 se representa el enfoque basado en reglas.

2.4.3 Enfoque basado en aprendizaje

Este enfoque proporciona al sistema clasificador la habilidad de aprender de manera automática y mejorar su desempeño con base en la experiencia adquirida en el proceso. Los

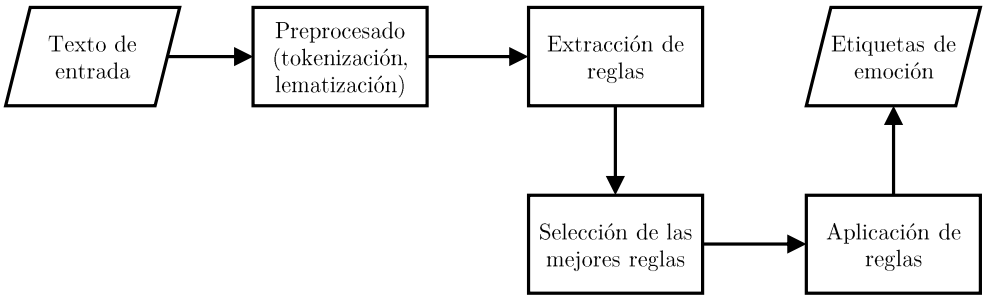


Figura 2.4: Enfoque basado en reglas

algoritmos de *Machine Learning* son usualmente utilizados en la clasificación mediante las técnicas de aprendizaje supervisado y no supervisado. Uno de los algoritmos comunmente frecuentados en la diferenciación de emociones en textos es el algoritmo de Máquinas de Vectores de Soporte por sus siglas en inglés SVM el cual es un algoritmo del tipo supervisado. En la figura 2.5 se representa el enfoque basado en aprendizaje.

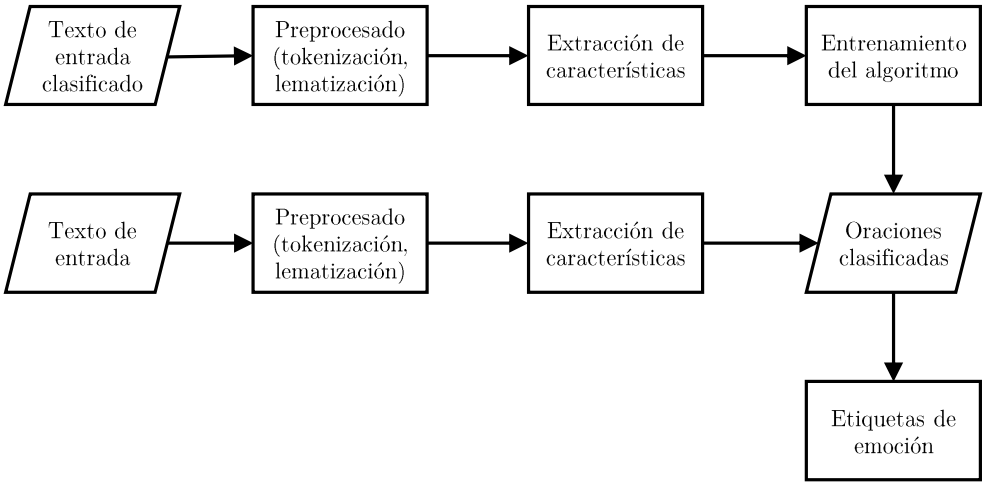


Figura 2.5: Enfoque basado en aprendizaje

2.4.4 Enfoque basado en aprendizaje profundo

El aprendizaje profundo es una rama del *Machine Learning* en donde los diferentes programas implementados manejan el concepto de aprender de la experiencia donde el mundo es interpretado en conceptos de jerarquía de conceptos; realizando la definición de cada uno de los conceptos en términos de la relación con conceptos más sencillos. Esta aproximación permite que un programa adquiera conceptos complicados realizando su construcción a partir de conceptos más simples [27]. Dentro de los modelos de aprendizaje profundo más utilizados se encuentra el de *Long Short Term Memory* (LSTM), el cual consiste en una forma especial de red neuronal recurrente con la capacidad de manejar dependencias a largo plazo. Este método cuenta con la ventaja que supera medianamente el problema de desvanecimiento del gradiente. En la figura 2.6 se representa el enfoque basado en aprendizaje profundo.

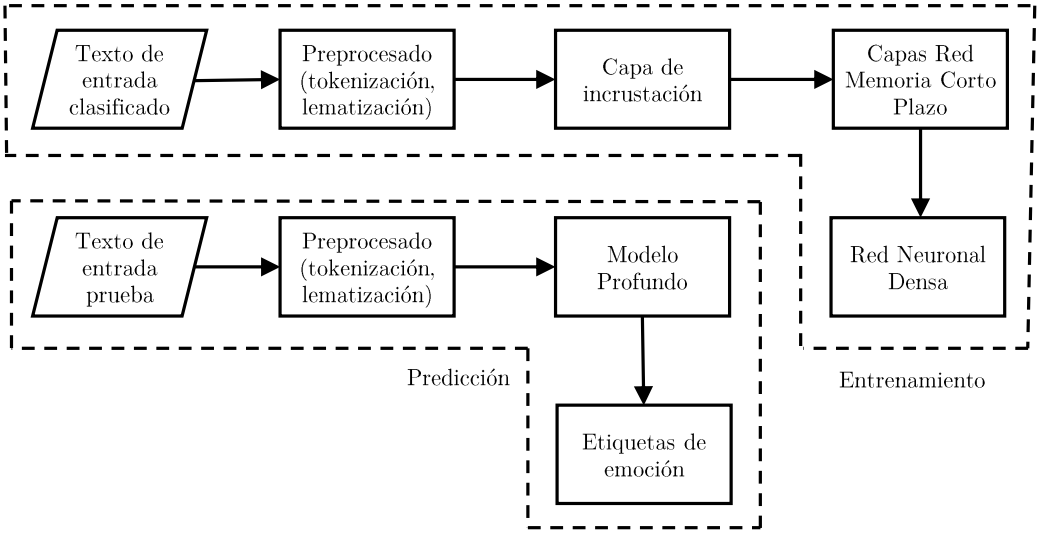


Figura 2.6: Enfoque basado en aprendizaje profundo

2.4.5 Medidas de evaluación

Cada uno de los trabajos identificados en el estado del arte realizan la medición de su desempeño mediante diferentes técnicas, para así realizar la medición de la similitud entre los conjuntos pronosticados contra los datos reales y también identificar el nivel de precisión en el pronóstico. Algunas medidas identificadas en los documentos son las siguientes.

Índice de Jaccard

$$\mathbf{IJ} = \frac{1}{|S|} \sum_{s \in S} \frac{|G_s \cap P_s|}{|G_s \cup P_s|} \tag{2.1}$$

Donde S es el conjunto de frases a clasificar, G_s el conjunto de todas las etiquetas disponibles para clasificación y P_s el conjunto de todas las etiquetas predichas para la frase.

Precisión

$$\mathbf{P_e} = \frac{VP_e}{VP_e + FP_e} \tag{2.2}$$

Donde e corresponde a la etiqueta de emoción, VP a Verdadero Positivo y FP a Falso Positivo.

Recall

$$\mathbf{R_e} = \frac{VP_e}{VP_e + FN_e} \tag{2.3}$$

Donde e corresponde a la etiqueta de emoción, VP a Verdadero Positivo y FN a Falso Negativo.

F_{score}

$$\mathbf{F_1} = 2 \times \frac{P_e \times R_e}{P_e + R_e} \tag{2.4}$$

P_e corresponde al valor de la *precision* y R_e al valor del *recall* calculados previamente.

Accuracy

$$\mathbf{A} = \frac{\sum_{e \in E} VP + \sum_{e \in E} VN}{\sum_{e \in E} VP + \sum_{e \in E} VN + \sum_{e \in E} FP + \sum_{e \in E} FN} \tag{2.5}$$

Donde E corresponde al conjunto de etiquetas de emoción, VP corresponde al número de verdaderos positivos, VN es el número de verdaderos negativos, FP es el número de Falsos Positivos y FN es el número de falsos negativos.

2.5 Transformes: La atención en el *machine learning*

Desde la publicación del primer *paper* a finales del año 2017, por parte de Google, [1] se realizó por primera vez la introducción a la arquitectura del *transformer*, caracterizándose por ser un modelo en el cual se realiza el reemplazo de las capas recurrentes de los modelos predominantes por las nuevas y llamadas capas de atención. El objetivo principal de estas capas de manera inicial consiste en realizar la codificación de cada una de las palabras presentes en un texto y con base al resto de las palabras consecuentes se realiza así la inclusión del contexto de las diferentes frases que componen el texto y realizar la representación matemática correspondiente.

Las Redes Neuronales Recurrentes (RNN) son populares y altamente utilizadas en las tareas de procesamiento de lenguaje natural; pero poseen un gran problema: son de la denominada memoria a corto plazo y suele ser un inconveniente considerable en aquellas tareas donde hay una frase de entrada bastante extensa y se presenta una relación fuerte entre palabras que se encuentran al inicio de la frase con palabras al final; por lo que dentro del modelo aquellas palabras que se hallan al final de la cadena de entrada van a representar un mayor peso en la predicción final del modelo. Para analizar más a fondo este inconveniente a continuación se presenta con más detalle cada una de las partes.

2.5.1 Codificadores, decodificadores y aprendizaje secuencia a secuencia

Antes de las arquitecturas de atención, eran común los modelos de secuencia a secuencia (*Seq2Seq*) donde los elementos de entrada son ingresados de manera consecutiva a la capa codificador, generando su representación intermedia y finalmente esta representación pasa por un decodificador para así generar finalmente una predicción de una secuencia final. en resumen, el objetivo consiste en realizar la transformación de una secuencia de entrada en una nueva como se evidencia en la figura 2.7, donde el valor de x es la secuencia de entrada, x_1, x_2, x_n los token de la secuencia, y a la secuencia de salida y y_1, y_2, y_n los token de la secuencia de salida.

Las arquitecturas de codificador y decodificador representan en su interior varias capas de Redes Neuronales Recurrentes apiladas donde el codificador procesa la entrada, genera una representación compacta de cada uno de los elementos de entrada en un momento diferente en el tiempo de manera secuencial, donde la salida de cada una de las capas hace una retroalimentación adicional para el segundo elemento de la entrada como se

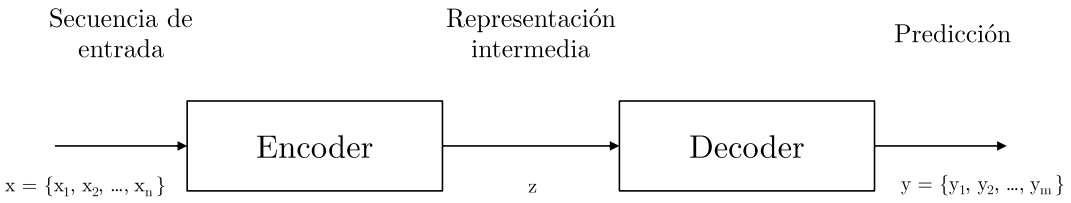


Figura 2.7: modelo secuencia a secuencia

evidencia en la figura 2.8 sección izquierda. Y por el otro lado tenemos el decodificador que recibe como entrada el vector de contexto z y genera la secuencia de salida en forma de predicción como se evidencia en la figura 2.8 sección derecha. Cabe resaltar que la predicción es realizada de manera secuencial.

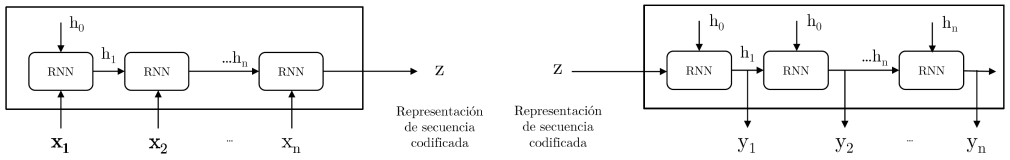


Figura 2.8: Modelo general de decodificador

El modelo funciona eficientemente únicamente con secuencias de texto que no superen 20 intervalos de tiempo (o tokens para el caso de procesamiento de frases). Las representaciones intermedias z no pueden realizar una codificación de todos los pasos en el tiempo. Este problema se conoce como de cuello de botella ya que el vector z necesita contar con toda la información que se pueda obtener de la secuencia de entrada, por lo que se puede decir de manera coloquial que las RNN tienden a olvidar la información de aquellos pasos que encuentran más al inicio del tiempo de procesamiento en la secuencia de entrada.

Si consideramos un párrafo de un texto que usualmente llega a tener entre 100 palabras, pueden existir relaciones importantes entre la palabra número 2 y la palabra número 95 del texto en mención. En la mayoría de los casos el vector z no podrá comprimir la información de la palabra 2 y la palabra 95 ya que el sistema enfocaría su atención a la última palabra relevante. Esta situación es ejemplificada con el problema de desvanecimiento del gradiente ilustrado en la figura 2.9.

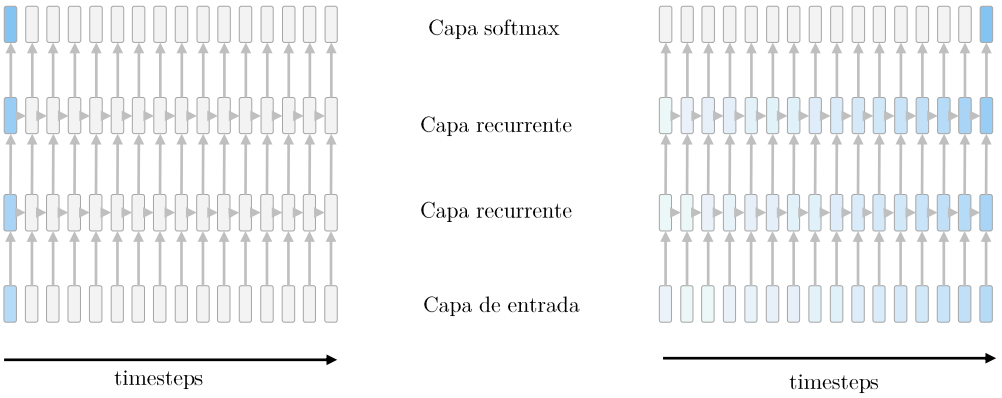


Figura 2.9: Problema desvanecimiento del gradiente

2.5.2 La atención

Los modelos de atención nacen como la alternativa para solucionar el inconveniente de pérdida de memoria en el transcurso del tiempo estableciendo una conexión directa entre cada uno de los elementos en los diferentes intervalos. La idea inicial surgió de los modelos de visión por computadora en los que al analizar diferentes partes de una imagen se puede llegar a obtener información acumulada sobre alguna forma en particular y de acuerdo con esta poder realizar la clasificación correcta de la imagen.

Este principio fue extendido posteriormente a las secuencias de texto, en donde al tomar todas las palabras que lo conforman, el objetivo consiste en poner atención a aquellas relaciones presentes entre todas las palabras del conjunto.

En la literatura se definen dos tipos de atención, suave y fuerte. Para ambos casos, la atención es configurada por funciones diferenciales. Cuando la función varía de forma suave sobre los elementos del dominio y el resultado es diferenciable, se hace referencia a la atención suave la cual toma valores continuos en sus resultados. Por otro lado, el aprendizaje fuerte se genera cuando la función que la define toma valores discretos, es decir, es un mecanismo que indica si la atención debe prestarse sobre una región o no del sistema por lo que esta tiene cambios abruptos en su dominio.

Representación de los datos de entrada

Para realizar el ingreso de datos a la arquitectura de transformadores, la idea de cómo alimentar el modelo surge de la premisa de realizar la alimentación del modelo con toda la secuencia posible sin dependencias entre estados ocultos. Por ejemplo, para la frase *“Yo*

leo muchos libros", tendremos los siguientes elementos de entrada al modelo

I = {{"Yo","leo","muchos","libros"},
{"muchos","libros","yo","leo"},
{"leo","libros","yo","muchos"}...}

Este proceso es denominado tokenización y en vez de entregar una secuencia de elementos como sucedería comúnmente, entrega un conjunto de elementos distintos entre sí donde la disposición de cada uno de los elementos en el conjunto es irrelevante.

Word Embeddings

Consiste en realizar la representación en un espacio vectorial con valores de tipo continuos de acuerdo a alguna característica en particular a extraer. Debido a que las palabras al representar correlaciones entre ellas existen múltiple asociaciones entre las mismas por lo que dependiendo de la tarea de procesamiento a realizar, las palabras pueden estar "cerca" o "lejos" según sus características semánticas. En la figura 2.10 se representa esta asociación.

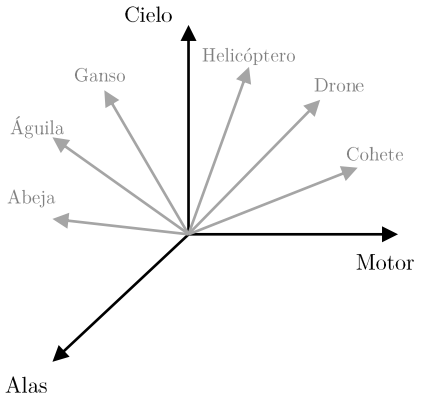


Figura 2.10: Similitud entre palabras y dimensiones

Codificación posicional

Al realizar la conversión de la secuencia de palabras se pierde la relación de orden entre ellas, por lo que el primer paso en la arquitectura transformador consiste en dar un sentido de orden a las mismas, realizando una modificación ligera de las representaciones vectori-

ales en función de la posición. En resumen, la codificación posicional consiste en realizar el añadido de constantes que se incluyen al vector antes de que ingresen a la primera capa de atención de la arquitectura. Por lo que si una misma palabra aparece en una posición diferente, la representación del vector va a variar dependiendo de dónde aparece la palabra en el insumo de entrada inicial. Los autores del documento original realizan la propuesta de la función sinusoidal como forma de realizar ésta codificación posicional. La función seno le indica al modelo que se debe prestar atención a una longitud de onda λ. Dada una señal $y(x) = \sin kx$ la longitud de onda será $k = \frac{2\pi}{\lambda}$. En este caso, el valor de λ será dependiente en la posición en la oración de entrada. i es utilizado para distinguir entre posiciones pares e impares. De esta forma se generan las siguientes ecuaciones:

$$\mathbf{PE}_{(\mathbf{pos},\mathbf{2i})} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{modelo}}}}\right) \tag{2.6}$$

$$\mathbf{PE}_{(\mathbf{pos},\mathbf{2*i+1})} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{modelo}}}}\right) \tag{2.7}$$

Capa de Auto-Atención

La auto-atención es el mecanismo de atención que relaciona diferentes posiciones de una única secuencia con el objetivo de realizar el cálculo de una representación asociada a la secuencia original. Esto permite establecer y encontrar correlaciones entre diferentes palabras de la entrada indicando la estructura sintáctica y contextual de la oración. Tomando la frase de ejemplo, “Yo leo muchos libros”, una capa entrenada de auto-atención realizará la asociación de la palabra “Leo” con las palabras “Yo” y “Libros” con un mayor peso que la palabra “muchos”. Lingüísticamente se sabe que estas palabras componen una relación de tipo sujeto-verbo-objeto y se puede deducir de manera intuitiva lo que la capa de auto-atención generará.

Los transformadores realizan el uso de tres objetos para estas representaciones, las consultas (Q), las llaves (K) y los valores de la matriz del espacio vectorial (V). Esto se realiza de manera sencilla realizando la multiplicación de la entrada $X \in R^{N \times dk}$ con tres matrices de pesos diferentes denominadas W_Q, W_K y $W_V \in R^{dk \times d_{modelo}}$. Básicamente consiste, en una multiplicación matricial sobre el espacio vectorial inicial. La dimensión resultante va a ser de un tamaño menor $d_k > d_{modelo}$. Contando ya con las matrices indicadas anteriormente, se puede realiza la aplicación de la capa de auto-atención con la

siguiente fórmula:

$$\textbf{Atención}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.8}$$

La función de producto escalar es la seleccionada como función para realizar la representación de la correlación entre dos palabras. El término de la función encuentra la similitud de la consulta con una búsqueda, para así finalmente realizar la aplicación de una función *softmax* para obtener los resultados finales de la atención como una distribución de probabilidad.

Capa de Normalización

En estas capas, la media y la varianza son calculadas de acuerdo a los canales y las dimensiones espaciales. Como cada una de las palabras es un vector, solo hay una única dimensión con las cuales se tratan los vectores. La capa de normalización realiza uso de las siguientes ecuaciones:

$$\mu_{\mathbf{n}} = \frac{1}{K}\sum_{k=1}^{K} x_{nk} \tag{2.9}$$

$$\sigma_{\mathbf{n}}^{\mathbf{2}} = \frac{1}{K}\sum_{k=1}^{K}(x_{nk} - \mu_n)^2 \tag{2.10}$$

$$\hat{\mathbf{x}}_{\mathbf{nk}} = \frac{x_{nk} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}, \hat{x}_{nk} \in R \tag{2.11}$$

$$\ln \gamma, \beta \mathbf{x_n} = \gamma \hat{x}_n + \beta, x_n \in R^K \tag{2.12}$$

Capa de transformación lineal

Se realiza mediante la siguiente ecuación:

$$\mathbf{y} = xW^T + b \tag{2.13}$$

Donde W es una matriz, y x, y, b son vectores. En la arquitectura se añaden dos capas lineales.

El núcleo del proceso: atención multi-cabeza e implementación en paralelo

En el documento de Google [1], se muestra la idea de pasar de un esquema de auto-atención a uno de atención con múltiples cabezas donde el objetivo consiste en realizar la aplicación del mecanismo de atención múltiples veces. En cada una de las iteraciones se realiza un mapeo independiente del conjunto de los tres objetos vistos anteriormente (matrices de llave, consulta y valor) desarrollando el proceso en diferentes espacios dimensionales y se realiza el cálculo de la atención en cada una de las dimensiones. A éste resultado se le llama "cabeza". El mapeo es obtenido realizando la multiplicación de cada una de las matrices con una matriz independiente de pesos, denotada como $W_i^K, W_i^Q \in R^{d_{modelo} \times dk}$, y $W_i^V \in R^{d_{modelo} \times dk}$.

Para realizar una compensación por la complejidad adicional, el vector de salida es dividido por el número de cabezas generadas. Las cabezas son concatenadas y transformadas utilizando una matriz de pesos cuadráticos $W^O \in R^{d_{modelo} \times d_{modelo}}$, teniendo en cuenta que $d_{modelo} = hdk$. Juntando todo se obtiene la siguiente ecuación:

$$\textbf{MultiCabezas}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = concat(cabeza_1, ..., cabeza_h)W^O \tag{2.14}$$

$$donde, cabeza_i = Atención(QW_i^Q, KW_i^K, VW_i^V) \tag{2.15}$$

La atención de múltiples cabezas es la que permite atender diferentes partes de la secuencia de una forma diferente por cada una de las veces que se procesa, esto significa que el modelo puede realizar la captura de información posicional ya que cada una de las cabezas va a procesar diferentes segmentos de la entrada. La variedad de combinaciones representan de manera más robusta el modelo. Y adicionalmente cada una de las cabezas obtendrá información contextual diferente realizando correlaciones entre las palabras de una manera distinta.

El codificador en los transformadores

Resumiendo lo evidenciado en las secciones anteriores, para procesar una frase se requieren tres pasos iniciando con la construcción del espacio vectorial del texto de entrada el cual es calculado de manera simultanea. Posteriormente la codificación posicional es aplicada a cada uno de los resultados en vectores de palabras para realizar la inclusión de información posicional y por útimo, los vectores resultantes son entregados al primer bloque codificador.

Cada uno de los bloques codificadores se componen de capas ordenadas iniciando con una capa de auto atención con múltiples cabezas para encontrar la correlación entre cada una de las palabras, una capa de normalización, una capa de conexión residual alrededor de las anteriores dos subcapas, una capa lineal, segunda capa de normalización y se finaliza con segunda conexión residual. Cada uno de los bloques puede ser replicado múltiples veces para así formar la estructura final del codificador. En [1] se propone un modelo con 6 bloques.

El decodificador en los transformadores

El decodificador consiste en la inclusión de los mismos componentes del codificador con al inclusión de dos nuevos. La secuencia de salida es alimentada en su totalidad y se calculan los vectores de espacio temporal. La codificación posicional se aplica de nuevo y los vectores son entregados al primer bloque decodificador.

Cada uno de los bloques decodificadores está compuesto por una capa enmascarada de auto-atención de múltiples cabezas, una capa de normalización seguida de una conexión residual seguida de una nueva capa de atención con múltiples cabezas (conocida como capa de atención codificador-decodificador), una segunda capa de normalización y una conexión residual y finalmente una capa lineal y una tercera conexión residual

El bloque de decodificación aparece de igual manera 6 veces. La salida final es transformada mediante una última capa lineal y las funciones de distribución de probabilidad son calculadas con la función conocida *softmax*. Las probabilidades de salida realizan la predicción del siguiente token en la frase de salida. Realizando la unión de todos los bloques expuestos anteriormente, se cuenta con la totalidad de la arquitectura del *transformer* que es la evidenciada en la figura 2.11.

2.6 Procesamiento de lenguaje natural y modelos de lenguaje

El procesamiento de lenguaje natural es considerado como uno de los campos de mayor relevancia en de investigación debido a los diferentes avances en los nuevos modelos de aprendizaje profundo. Su objetivo principal consiste en la generación de modelos mediante diferentes técnicas de aprendizaje máquina, para así, llegar a cierto nivel de comprensión del lenguaje humano. Dentro de las principales tareas en las que se utilizan las tareas de procesamiento de lenguaje natural se encuentran la comprensión de documentos [28],

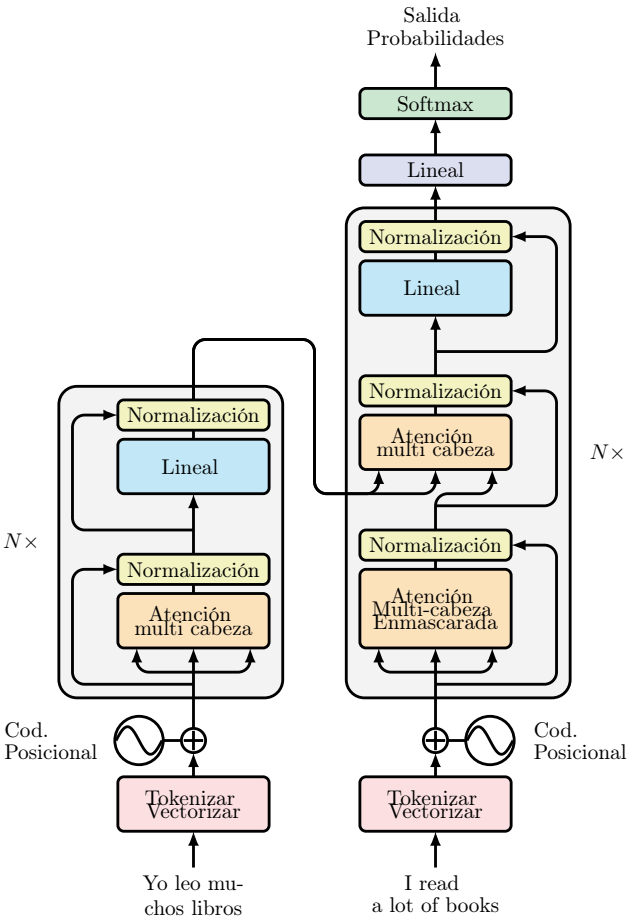


Figura 2.11: Arquitectura general del modelo *transformer* (adaptado de [1]).

reconocimiento de voz [29], recuperación de información [30], elaboración inteligente de textos [31], respuesta a preguntas [32], análisis de sentimientos [33], entre otras.

El lenguaje natural implica el análisis y el tratamiento de una gran número de palabras que pueden introducir una amplia variedad de ambigüedades en los textos a analizar. La ambigüedad léxica donde una palabra puede tener varios significados, la ambigüedad sintáctica donde una oración puede contar con múltiples ramas, o la ambigüedad anafórica donde una una frase a la que hace referencia una palabra puede tener numerosos resultados. Debido a que las máquinas no pueden procesar las ambigüedades de la misma manera en la que la realizamos los seres humanos, las técnicas de procesamiento de lenguaje natural en este caso realizan el cálculo de una valor de probabilidad asociado a una secuencia de tokens que tiene una ocurrencia dentro de un corpus [30]. De esta manera, se obtiene un modelo probabilístico que ayuda a realizar la predicción de una palabra con base en una

secuencia de ellas para así generar un resultado final.

De allí surgen los denominados modelos de lenguaje, que surgen con el objetivo de realizar la predicción de secuencias de tokens con base en la información utilizada para realizar el entrenamiento del modelo. De esta forma, se realiza el diseño del modelo con el dominio de los datos ingresados. Estos datos deben representar el comportamiento real con la mayor fidelidad posible para que de esta manera, el resultado generado por los modelos entrenados sea lo más aproximado al comportamiento del mundo real. La intuición lingüística desempeña un papel fundamental en la implementación de estos modelos, debido a que es el el elemento que almacena el conocimiento implícito adquirido con los datos proporcionados y con los que ha sido generado el modelo. Este genera juicios sobre si una oración se encuentra sintácticamente correcta, sinonimia entre palabras o inclusive la forma en que suelen ser finalizadas las oraciones. Para el presente trabajo se realizó el uso de dos modelos de lenguajes diferentes. Para la fase de detección de tendencias suicidas se realizó uso del modelo BERT, y para la frase de transformación y validación de texto *leet* a texto en claro, fue utilizado el modelo GPT2.

2.6.1 BERT (*Bidirectional Encoder Representations from Transformers*)

De acuerdo con el documento publicado por sus desarrolladores en Google en el año 2018 [2], BERT realiza uso de los mecanismos de atención para así obtener una relación contextual entre las diferentes palabras que componen un archivo textual. Como se comentó anteriormente, la arquitectura de los transformadores utiliza un codificador y un decodificador, pero debido a que el objetivo principal de BERT consiste únicamente en general el modelo de lenguaje, sólo el componente codificador es utilizado.

La ventaja de BERT sobre los modelos direccionales, es realizar la lectura de la secuencia de texto de manera completa de todas las palabras que la componen, por ello se denomina bidireccional. De esta manera, es posible que el modelo adquiera el contexto de una palabra en función del entorno que la rodea (palabras a su izquierda y derecha).

En la figura 2.12, se representa en alto nivel el funcionamiento de codificación de BERT, donde la entrada consiste en una secuencia de tokens, que son integrados en vectores y posteriormente procesados por la red neuronal. La salida es una secuencia de vectores de tamaño H, en la que cada vector corresponde a un token de entrada con el mismo índice.

Al realizar el entrenamiento de modelos de lenguaje, existe el desafío de definir un

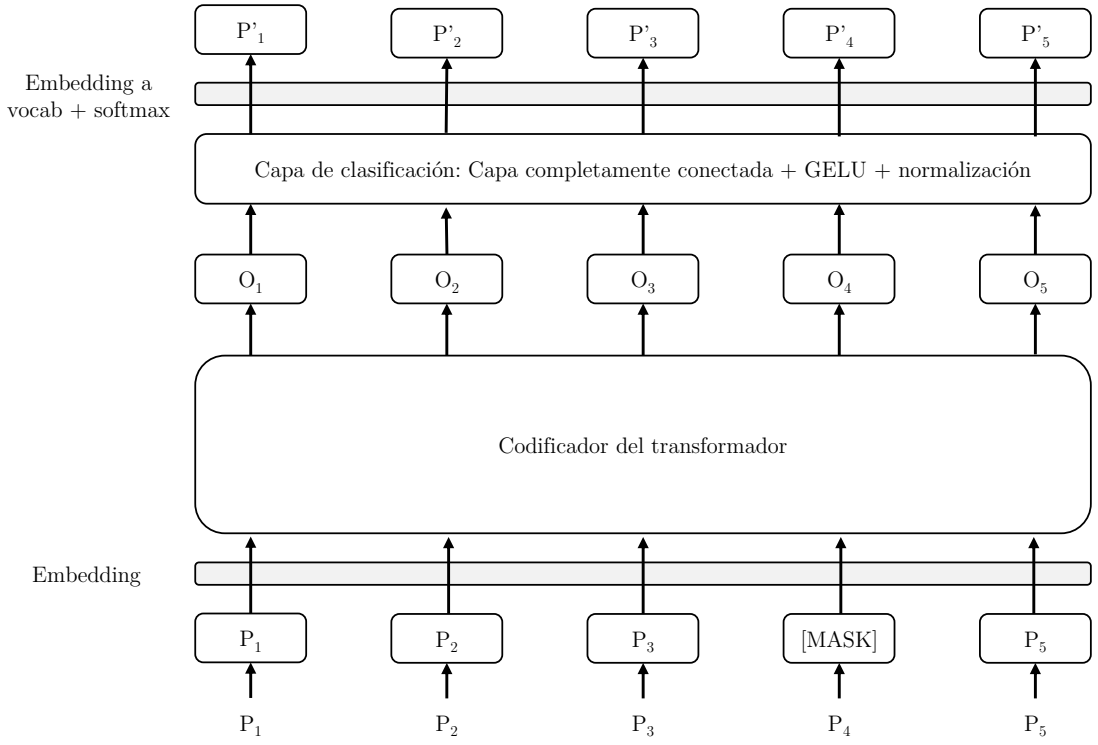


Figura 2.12: Arquitectura general del modelo de lenguaje BERT (adaptado de [2]).

objetivo de predicción. Muchos modelos predicen la siguiente palabra en una secuencia, un enfoque direccional que limita el aprendizaje sobre el contexto. Para prevenir esta limitante, BERT utiliza dos estrategias de entrenamiento.

Modelo de lenguaje enmascarado (MLM)

Previo a la introducción de las secuencias de palabras, el 15% de las palabras de la secuencia total del texto son enmascaradas con un token del tipo "[MASK]". Posteriormente, el modelo intenta realizar la predicción el valor original de las palabras que han sufrido este enmascaramiento con base al contexto que han proporcionado las demás palabras de la secuencia. Técnicamente, esta predicción requiere de la inclusión de una capa de clasificación a la salida del codificador. Realizar la multiplicación de los vectores de salida por la matriz de *embeddings* transformándolos así en la dimensión del vocabulario. Y por último, realizar el cálculo de la probabilidad de cada palabra en el vocabulario con una función *softmax*.

La función de pérdida del modelo solo tiene en cuenta la predicción de los valores que han sido enmascarados ignorando las probabilidades de las palabras no enmascaradas.

De esta manera el modelo tiene una convergencia más lenta con respecto a los modelos direccionales, pero esto se compensa con un mejor desempeño al detectar el contexto.

Predicción de la siguiente oración (NSP)

En el proceso de entrenamiento, el modelo recibe parejas de oraciones como entrada y aprende a predecir si la segunda oración del par corresponde a la siguiente oración del documento original. En el entrenamiento, el 50% de las parejas de entrada son un par en el que la segunda frase es la siguiente en el documento, mientras que en el restante 50% se toma una palabra al azar del corpus como segunda frase. La presunción consiste en que la oración seleccionada aleatoriamente no tiene ningún tipo de conexión con la primera.

Para realizar la distinción entre dos oraciones en el proceso de entrenamiento, se realiza un preprocesamiento iniciando con la inserción de un token "[CLS]" al inicio de la primera oración y un token "[SEP]" al final. A cada uno de los token se añade una indicación a cada token identificando si pertenece a la oración A o la oración B. Y finalmente, se agrega una incrustación posicional a cada uno de los tokens para indicar la posición en la secuencia. Este proceso se muestra en la figura 2.13.

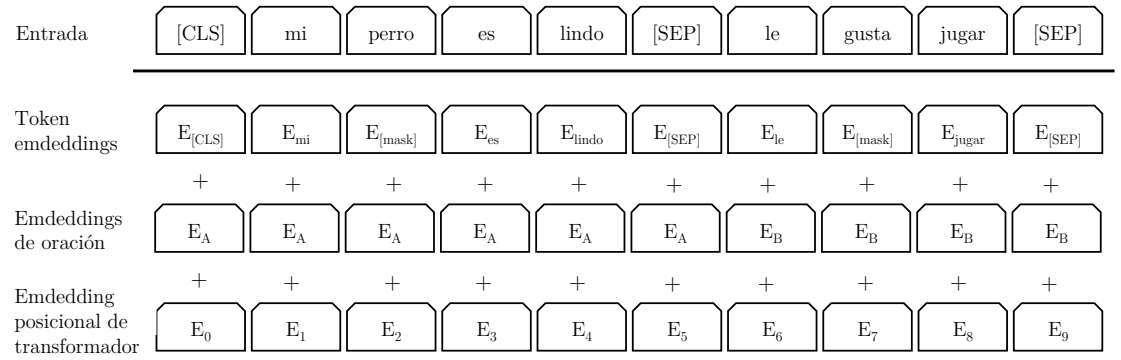


Figura 2.13: Arquitectura general del modelo para predicción de siguiente oración [2]).

Finalmente, para predecir si la segunda oración se encuentra asociada con la primera, en primer lugar se ingresa la secuencia completa al modelo transformador. Luego, la salida del token "[CLS]" es transformada en un vector de 2×1, usando una capa simple de clasificación mediante matrices aprendidas de pesos y biases. Y finalmente, se realiza el cálculo de la probabilidad del elemento denominado *IsNextSequence* mediante *softmax*.

Al entrenarse el modelo BERT, MLM y NSP son entrenados de manera simultanea,

con el objetivo de minimizar la función de pérdida combinada de las dos estrategias implementadas.

2.6.2 Familia GPT2

Modelo *DistilGPT2*

Su nombre es una abreviatura para *Distilled-GPT2*. Corresponde a la versión más pequeña, rápida y simplificada del GPT2 y utiliza la arquitectura BERT. Cuenta con 6 capas, 768 dimensiones y 12 cabezas, lo que totaliza un total de 82 millones de parámetros entrenados. Para su proceso de entrenamiento se manejó el conjunto de datos *OpenWebText*; que incluye un total de 8013769 documentos con más de 128 palabras cada uno y son documentos subidos por miles de usuarios a la plataforma *reddit*. El modelo tardó un total de 168 horas de entrenamiento y fue ejecutado completamente en la nube.

Modelo *GPT2*

La primera versión del modelo nacido en el año 2019. Cuenta con un total de 124 millones de parámetros. Su entrenamiento fue implementado de igual manera con el conjunto de datos *OpenWebText*, con la diferencia que para este modelo fueron excluidas todas las fuentes de Wikipedia. No se han hecho públicos los documentos sobre los que se realizó el entrenamiento, pero si se encuentra un listado con los primeros 1000 sitios de donde fueron extraídos, organizados por la cantidad de documentos de cada uno. El tamaño de su vocabulario es de 50257 palabras y el contenido de cada frase es de 1000 palabras. Fue entrenado en 256 núcleos de TPU en la nube versión 3, pero no se cuenta con el tiempo total de entrenamiento.

Modelo *GPT2-Medium*

Es el modelo más grande de los tres seleccionados. Incluye un total de 355 millones de parámetros y fue entrenado con el objetivo de modelado de lenguaje casual. De igual manera, fue utilizado el conjunto de datos *OpenWebText* y de forma similar se descartaron los documentos provenientes de Wikipedia. Su entrenamiento se efectuó de manera autosupervisada, es decir, sin que los textos utilizados en su entrenamiento fueran etiquetados con anterioridad. El tamaño de su vocabulario es de 50257 palabras y el tamaño de cada frase es de 1000 palabras. El objetivo de los tres modelos GPT-2 es tratar de predecir la

palabra siguiente en un conjunto de oraciones.

Debido a que los modelos de lenguaje por la cantidad de palabras y combinaciones de las mismas genera predicciones con valores de probabilidad bastante pequeños, se hizo necesario utilizar una escala que representara de una mejor manera las probabilidades obtenidas en cada una de las frases. Es por ello que *lm-scorer* permite seleccionar el formato en que son retornadas las probabilidades para cada una de las frases. Para esta implementación se hizo uso de la forma logarítmica de la probabilidad que computacionalmente es menos costosa al ejecutar operaciones tipo suma que multiplicaciones, para efectuar el cálculo final de la probabilidad en cada una de las oraciones. Además el uso de esta forma de probabilidad mejora el concepto de estabilidad numérica al manejar probabilidades tan pequeñas por la forma en la que los sistemas computaciones realizan la aproximación de los números reales. En la tabla 2.3 se muestran los sistemas de puntuación utilizando el sistema de probabilidad clásico contra la representación logarítmica.

Tabla 2.3: Puntuación de frase en representación probabilística clásica y logarítmica

Frase: *Testing the language model scoring*

Palabra	Probabilidad clásica	Probabilidad logarítmica
Testing	4.7485×10^{-5}	-9.9551
the	0.064932	-2.7344
language	0.00067324	-7.3034
model	0.00083548	-7.0875
scoring	1.7122×10^{-6}	-13.278
fin de frase	0.00079546	-7.1366
Total:	2.3621×10^{-21}	-47.495

Capítulo 3

Estado del arte

En el trabajo de [34], el objetivo principal consiste en realizar una predicción de manera anticipada de casos de depresión en las publicaciones realizadas por los usuarios. Como preprocesamiento del texto deciden por abandonar las técnicas comunes como las bolsas de palabras y realizar el modelado de los datos al igual que los modelos de predicción mediante el uso de grafos y clasificar las publicaciones en dos clases de resultados, si la persona que la escribió se encuentra deprimida o no deprimida. Las ejecuciones del modelo se realizaron con 5 configuraciones diferentes: LyRA donde el modelo es construido mediante los términos como palabras independientes. LyRB construye el modelo mediante modelos de 3-gramas. LyRC es una configuración híbrida generando un vector de 2 x 4 x N extraídas de las características de las dos configuraciones anteriores. LyRD es un método de ensamblaje conservador que utiliza las salidas de las tres configuraciones anteriores y realiza la asignación de deprimido si las tres primeras configuraciones coinciden en asignar dicha categoría. Finalmente, LyRE es similar a la configuración anterior, pero donde la clasificación se realiza con base en la clasificación de la mayoría de los tres modelos iniciales en un esquema 2 de 3.

En la investigación de [35], la meta consiste en medir manifestaciones conductuales asociadas a la depresión con un total de 21 categorías de síntomas y actitudes que de acuerdo con estudios clínicos son características que poseen aquellas personas con algún nivel de depresión. Cada categoría es graduada con 4 o 5 frases para validar con cual se puede sentir identificado el paciente y se asigna un rango de 0 a 3 para reflejar el rango de la severidad asociada a cada una de las categorías. Como método de clasificación se utilizó el método de *Naïve Bayes*.

Los siguientes dos trabajos ya se enfocan en tendencias más fuertes que corresponden a la identificación de tendencias suicidas en personas que ya pueden contar con un nivel bastante profundo de depresión. [36] realiza la identificación de publicaciones con tendencias suicidas en la red social *Reddit*. La metodología implementada consistió en el diseño e implementación de un modelo híbrido de Redes Neuronales Convolucionales (CNN) y redes recurrentes del tipo *Long Short Term Memory* (LSTM), donde las entradas al sistema son procesadas por la capa LSTM y sus salidas son el insumo para la sección CNN. Como resultado del estudio se obtuvieron aquellas palabras que son muy frecuentes y presentan una alta ocurrencia en aquellas publicaciones con tendencias suicidas como los son "suicidio", "quiero morir", "morir maldita sea", "deseo suicida", "deseo morir", "me quiero ir" entre otras. La medida de *accuracy* para esta implementación fue del 91.7%.

[37] de igual manera, realiza el análisis de tendencias suicidas en la red social *twitter* mediante un conjunto de tuits en idioma inglés, pero basando la recopilación de datos en un usuario en específico y considerando su comportamiento de publicaciones en un intervalo de tiempo; para así poder detectar con mayor precisión si existe un patrón en cada una de las publicaciones y poder llegar así a considerar un punto de alerta máxima en la que el usuario pueda llegar a intentar a hacer algo en contra de su vida. Como metodología implementada, se utilizaron transformadores entrenados previamente, en particular el denominado BERT. En una primera capa de este modelo realiza la tokenización de todas las publicaciones históricas para así realizar la creación de un vector emocional inicial. Para la capa subsiguiente se realizó la implementación de capas *Time-Aware Long-Short Term Memory* (T-LSTM) debido a que precisamente el comportamiento del usuario a analizar puede ser cambiante en los intervalos de tiempo, por lo que este sistema resulta más eficaz para comparar datos anteriores en el tiempo con comportamientos actuales. La medida de *accuracy* para esta implementación fue del 85.1%.

[38] utiliza tres conjuntos de datos, notas escritas a mano por personas que ha cometido suicidio, notas de última voluntad de prisioneros condenados a muerte entre 1982 y 2017 en el estado de Texas, y notas neutrales. Los autores realizan la comparación de ambos conjuntos de datos comparando el número promedio de palabras que componen las diferentes notas y el número de palabras en cada una de las oraciones obteniendo como resultado que el número de palabras en las notas de suicido es mayor que en otro tipo; esto debido a que las personas tratan de transmitir la mayor cantidad de sentimientos posibles antes de cometer el acto de acabar con su vida. Posterior al análisis de cada una

de las palabras,términos como "salud mental", menciones a personas, inclusive la palabra "vida", presentan una alta frecuencia de aparición. Con respecto a las notas de última voluntad, términos como "Dios", "Jesucristo", y "corredor de la muerte" son aquella que cuenta con una mayor frecuencia de aparición. Como componente técnico en su trabajo, se utiliza un transformador RNN obteniendo como resultado promedio para las tres etiquetas de clasificación *precision* de 95.0%, *recall* de 94.9% y *F1-score* de 94.9%.

En [39] se utiliza como conjunto de datos los comentarios de diferentes usuarios en sus propias publicaciones y en las de otros usuarios obtenidas de diferentes foros en *Reddit*. Para el trabajo se definieron dos etiquetas para clasificar a los usuarios. "autolesionados" y "no autolesionados" diferenciados por los valores de 1 y 0 respectivamente. El dataset es generado por CLEF eRisk para su edición 2021. Como herramienta técnica se utiliza el modelo BERT para generar los *embeddings* de las palabras para luego utilizar las características extraídas y posteriormente complementar el modelo con un clasificador de regresión logística. Sus métricas de desempeño son *precision* de 44.1%, *recall* de 75.9% y *F1-score* de 55.8%.

En el trabajo de [40] se utilizan publicaciones de la red social *Reddit*. El conjunto de datos contiene un total de 3549 textos sugestivos suicidas. El subreddit utilizado para la extracción es denominado "vigilancia del suicidio". Es una comunidad especial en la red donde los usuarios comparten abiertamente ideas suicidas, intentos de suicidio, pero también aportes de otros usuarios para ayudar a los demás a deshacerse de estas ideas. Y para datos no asociados al suicido, se realizó la extracción de publicaciones asociados a otras categorías de la red. En este estudio el título de la publicación fue importante debido a que contiene un resumen muy aproximado sobre el contenido general de la publicación y brindar una primera clasificación de tendencia. Como modelos de lenguaje utilizados en la etapa de clasificación, se encuentran BERT, ALBERT, ROBERTa y XLNET. El que genera los mejores resultados es ROBERTa con *precision* de 92.67%, *recall* de 98.44% y *F1-score* de 95.47%.

En [41] se utiliza un conjunto de datos consistente en la colección de tuits basado en un *lexicon* de 143 frases asociadas al suicidio. Fue realizado un filtrado de manera manual sobre cada uno de los tuits, obteniendo así un total final de 34,306 publicaciones en total. Algunos de los tuits fueron publicados por los mismos usuarios. Los tuits fueron clasificados en dos categorías, "Intento suicida presente" e "Intento suicida ausente". Las etiquetas fueron asignadas por estudiantes de sicología clínica. Cabe resaltar que el etiquetado de

cada una de las publicaciones fue realizado a nivel de tuit individual y no por hilos de publicaciones. Finalmente, 3984 tuits clasificados con tendencia suicida fueron obtenidos. Como modelos para la clasificación, se utiliza un modelo propio de los investigadores denominado STATENet que se basa en una arquitectura de doble transformador, y BERT como el generador de los *embeddings*. Como resultados del modelo se obtienen un *Accuracy* de 85.1%, *recall* de 81.0% y *F1-score* de 79.9%.

En [42], realiza uso del mismo dataset del trabajo anterior, pero realiza la inclusión de modelos de atención temporal al modelo STATENet presentado anteriormente y evidenciando una leve mejora. Las métricas obtenidas para esta implementación son *Accuracy* de 85.6%, *recall* de 81.3% y *F1-score* de 80.4%.

Es válido indicar que en los trabajos presentados, ninguno hace referencia al procesamiento de publicaciones que tengan contenido con *leet speking*, lo que reafirma la importancia que tiene esta variable, pero que no es tenida en cuenta en los diferentes trabajos de investigación.

En el cuadro 3.1 se presenta un resumen del estado del arte expuesto.

Tabla 3.1: Resumen de trabajos sobre reconocimiento de tendencias suicidas en textos en las redes sociales.

Referencia	**Herramienta - Algoritmo**	**Dataset**	**Desempeño**	
			Medida	**Valor (%)**
Villatoro-Tello2017	Reglas	eRISK 2017	Precision	69.00
			Recall	92.00
			F1-score	64.00
Hosseini-Saravani2020	Naïve Bayes modificado	CLEF/eRsik	Precision	93.23
			Recall	90.90
			F1-score	82.75
Tadesse2020	CNN + LSTM	Reddit	Accuracy	91.70
Sawhney2020	BERT + T-LSTM	twitter	Accuracy	85.10
Zhang2021	Transformador + RNN	Notas físicas	Precision	95.00
			Recall	94.90
			F1-score	94.90
Qamar	BERT + Regresión logística	Reddit	Presicion	44.10
			Recall	75.90
			F1-score	55.80
Haque2020	ROBERTa	Reddit	Precision	92.67
			Recall	98.44
			F1-score	95.47
Ramit2021	STATENet	twitter	Accuracy	85.10
			Recall	81.00
			F1-score	79.90
Sawhney2021	STATENet + Atención temporal	twitter	Accuracy	85.60
			Recall	81.30
			F1-score	80.40

Parte III

Desarrollo

Capítulo 4

Metodología de trabajo

El desarrollo del presente trabajo se estructuró en un total de cinco etapas, esquematizadas de manera general en la figura 4.1. A continuación se detalla cada una de las fases.

4.1 Fase de entrenamiento

Se realizó el entrenamiento del modelo de aprendizaje automático con un conjunto de datos seleccionado para esta tarea, generando así, un prototipo el cual fue validado, probado y verificado. Las métricas de desempeño permitieron concluir si el modelo cumplió con la tarea inicial, brindando una correcta clasificación de los tuits, diferenciando aquellos que indican tendencias suicidas y aquellos que no. El componente resultante de esta etapa, fue un modelo entrenado que se respaldó debido a su función de objeto reutilizable en algunas de las fases siguientes. En la figura 4.2 se presenta el detalle del proceso ejecutado en esta fase.

Se utilizaron un total de 29.836 tuits clasificados previamente con las etiquetas de **"_suicidal_"** para aquellos clasificados con contenido asociado a tendencias suicidas, obteniendo un total de 4.568 tuits en esta categoría. Y la etiqueta **"_NotSuicidal_"**, para los tuits que no presentan este tipo de tendencia. Esta categoría cuenta con un total de 25.268 entradas.

El procesamiento general sobre los datos de entrada consistió inicialmente en la tokenización, se procedió con la eliminación de algunos caracteres especiales, dentro de los cuales se hallaron ; [;]; , ; ‘], incluir los separadores al inicio y final de cada una de las cadenas de texto, truncar cada uno de los tuits al máximo permitido para el modelo, mapear los tokens a cada uno de sus identificadores, crear las máscaras de atención y finalmente,

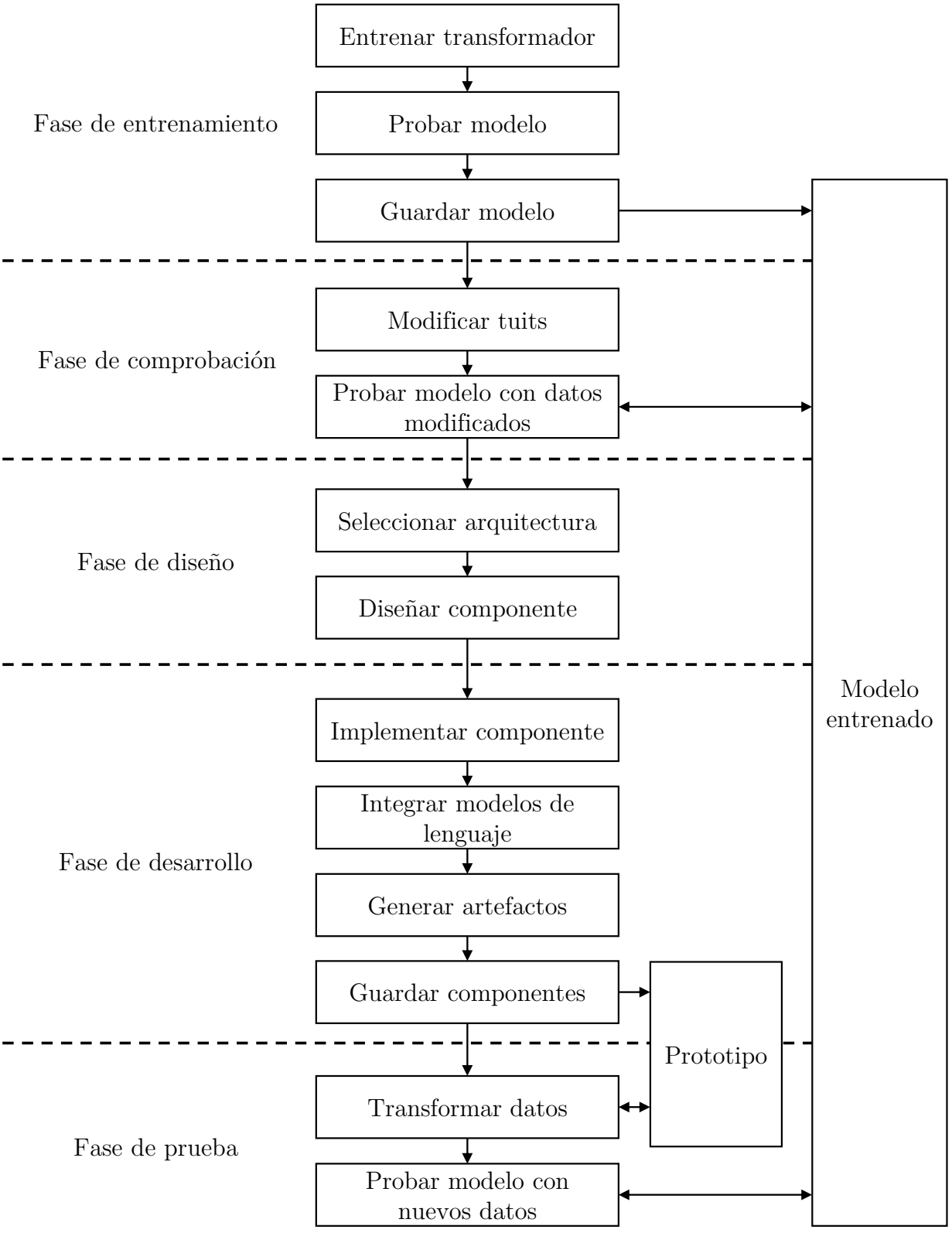


Figura 4.1: Representación del trabajo

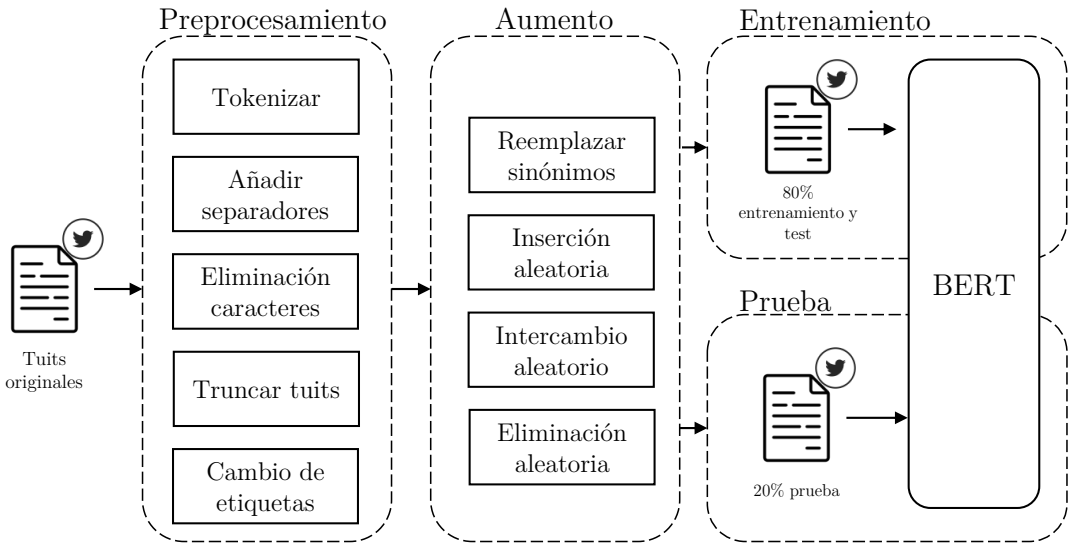


Figura 4.2: Proceso ejecutado en la fase de entrenamiento

retornar el resultado de este proceso.

Además, de haber convertido las etiquetas para cada uno de los tuits, de esta manera, aquellos con tendencias suicidas se asignaron a la etiqueta con valor "1"; y los no suicidas a la etiqueta con valor "0".

Los tuits marcados como suicidas, fueron inferiores en número a los clasificados como no suicidas. Es por ello que, la implementación, contó con el desarrollo de una función de aumento, incluyendo de este modo un mayor número de tuits clasificados dentro de la categoría de suicidas. El incremento fue efectuado con la codificación de cuatro funciones mediante el uso de librerías de procesamiento de lenguaje natural.

El primer método se denominó "reemplazo de sinónimos", cuya función fue seleccionar de manera aleatoria n número de palabras (que no sean gramaticales) de cada tuit, y posteriormente, fueron reemplazadas por sinónimos aleatorios extraídos del lexicón *WordNet*. La segunda función llamada "inserción aleatoria", de igual manera que la primera, buscó sinónimos aleatorios de n palabras que no sean gramaticales, y las insertó en posiciones aleatorias del tuit que se estaba procesando. La tercera función se denominó "intercambio aleatorio", y como lo indica su nombre, su objetivo fue tomar de manera aleatoria, dos palabras en el tuit e intercambiarlas. Y por último, la cuarta función, "eliminación aleatoria"; la cual suprimió una palabra del tuit que se procesó con un valor p de probabilidad el cual es calculado de manera aleatoria.

Finalizado el incremento, se obtuvo un total de 48089 tuits. 25268 clasificados como no suicidas, y un nuevo valor de 22.821 clasificados como suicidas. Posteriormente, se efectuó la división de los datos tomando el 80% del total, como datos de entrenamiento y validación; y el 20% de datos de prueba del modelo. Este último 20% fue también utilizado para aplicar la modificación *leet speaking* para así verificar el desempeño del modelo con las modificaciones experimentales que se efectuaron en la fase de comprobación.

Para este procedimiento se aplicó el modelo preentrenado denominado *bert-base-uncased* el cual no distingue palabras con letras mayúsculas o minúsculas, por lo que la palabra "Error" y "error" son consideradas como iguales. Este modelo, como lo indican sus desarrolladores, fue preentrenado en la nube con 4 TPU en configuración Pod para un total de 16 chips de TPU, con un total de un millón de pasos y tamaño de lote fue de 256. Las longitudes de las secuencias de entrada se limitan a 128 tokens para el 90% de los pasos y 512 para el 10% restante. El optimizador usado es el Adam con una tasa de aprendizaje de $0,0001$, $\beta_1 = 0,9$ y $\beta_2 = 0,999$.

Finalmente, se configuró el modelo BERT, las funciones para el optimizador, tasa de aprendizaje y el ciclo de entrenamiento. Este modelo con los tuits indicados, fue entrenado con un total de 4 épocas y 2062 lotes.

4.2 Fase de comprobación

Se seleccionó cerca del 20% de los datos utilizados en la fase de entrenamiento con un total de 9135 tuits; 4227 de estos corresponden a la etiqueta de tendencia suicida y 4908 clasificados como no suicida. Los 4227 tuits asociados a tendencias suicidas fueron modificados, con el objetivo de incluir diferentes escenarios de *leet speaking* en cada uno de los trinos que no fueron utilizados en el clasificador en su fase de entrenamiento. De esta manera, se comprobó el desempeño final con este insumo. Todos estos datos, pasaron por el proceso de tokenización del modelo BERT siendo inyectados a la fase de predicción tomando los datos y verificando el comportamiento. El proceso llevado a cabo se representa en la figura 4.4.

4.2.1 Modificación de palabras

Como primer escenario de pruebas, se utilizó la modificación de palabras clave solo en los tuits etiquetados dentro de la categoría de aquellos que incluyen tendencias suicidas,

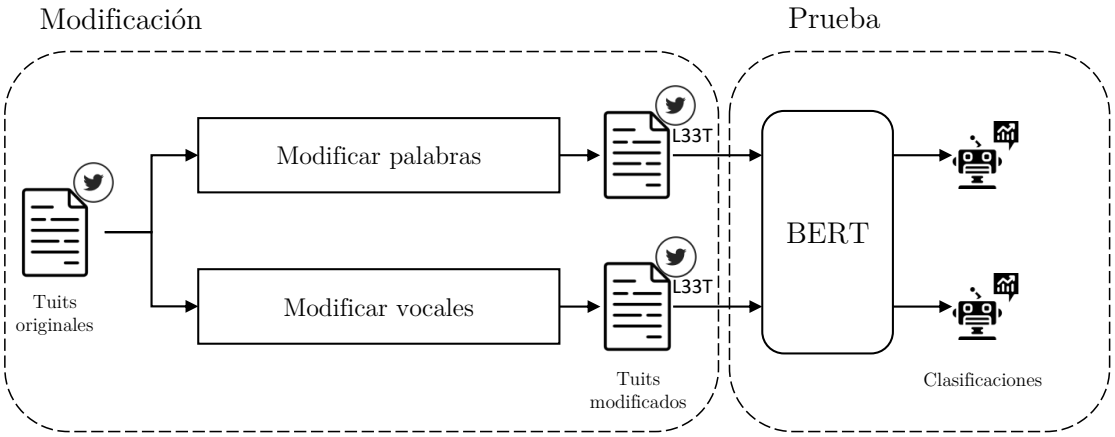


Figura 4.3: Proceso ejecutado en la fase de comprobación

los tuits clasificados como "No suicidas", no sufrieron ningún tipo de modificación. En cada uno de los experimentos, se ejecutaron los cambios descritos en el cuadro 4.1. Cada uno de los experimentos fue incremental, es decir, que las modificaciones implementadas para las primeras cuatro palabras fueron incluidas en los cambios para las siguientes cinco, para un total de nueve modificaciones y así sucesivamente. La prioridad en las palabras a modificar, fue definida con base en la frecuencia que aparecían en todo el conjunto de tuits, teniendo en cuenta qué, se descartaron las de categoría gramatical. Se determinó el cálculo de las frecuencias y fueron representadas en la figura 4.4 mediante dos nubes donde se compararon los tuits sin modificar y los tuits con la modificación de todas las 20 palabras.

Figura 4.4: Nubes de palabras entre tuits originales (izquierda) y tuits con veinte palabras modificadas

4.2.2 Modificación de vocales

Como segundo escenario, se propusieron seis experimentos adicionales; y la modificación, se realizó igualmente en los tuits clasificados de tipo tendencia suicida, cambiando la

Tabla 4.1: Modificación de palabras en claro a codificación *leet speaking*

Experimento	Palabra original	Palabra modificada
Grupo de palabras 1	depress	d3pr3ss
	really	r34ll1
	life	l1f3
	never	n3v3r
Grupo de palabras 2	work	w0rk
	help	h3lp
	still	st1ll
	much	mu(h
	suicid	su1c1d
Grupo de palabras 3	day	d4y
	sad	s4d
	peopl	p3opl
	think	th1nk
	time	t1m3
Grupo de palabras 4	one	0n3
	go	g0
	tell	t3ll
	kill	k111
	end	3nd
	need	n33d

totalidad de las vocales por uno de los caracteres equivalente en *leet* para cada vocal en todos los tuits. En cada uno de los experimentos, se incluyeron las modificaciones descritas en el cuadro 4.2. Los escenarios para esta etapa fueron independientes, es decir, las modificaciones utilizadas para la letra "a" no se acumularon con los de la letra "e" y así sucesivamente. Esto a excepción del último experimento al que se agregó la modificación de todas las vocales de manera simultánea. De igual manera, se efectuó el cálculo de las frecuencias representada en la figura 4.5 mediante dos nubes de palabras donde se compararon los datos con los tuits originales y los tuits con todas sus vocales modificadas.

Figura 4.5: Nubes de palabras entre los tuits originales (izquierda) y tuits con todas sus vocales modificadas (derecha)

Tabla 4.2: Detalle de modificación de vocales para tuits con tendencias suicidas

Experimento	Letra(s) Original(es)	Letra(s) modificada(s)
“a” por “4”	a	4
“e” por “3”	e	3
“i” por “1”	i	1
“o” por “0”	o	0
“u” por “v”	u	v
Todas las vocales	a, e, i, o ,u	4, 3, 1, 0, v

4.3 Fase de diseño

Para la fase de diseño, se procedió en primer lugar a determinar la arquitectura más conveniente, de modo que cada uno de los componentes que se desarrollaron, fueran modulares, independientes y desacoplados logrando así, que la modificación de uno de ellos no implicara cambios ni pruebas sobre los demás componentes interrelacionados. Es por ello que, la arquitectura orientada a microservicios fue la seleccionada como la mejor opción para la implementación de la solución. De esta manera, se dividió el módulo en componentes de software y cada uno de ellos se responsabilizó por resolver una fracción de problema de manera independiente, generando interfaces fijas como mecanismo de intercambio de información entre los módulos.

4.3.1 Módulo de conversión

En el módulo inicial, se realizó el proceso de tratamiento de cada uno de los tuits hasta llegar a nivel de carácter. De esta manera, se efectuaron las modificaciones respectivas obteniendo así el total de posibles frases por las que puede ser reemplazado un tuit.

En primer lugar, se modificaron todos los caracteres que pudieran aparecer en mayúsculas, obteniendo así, el equivalente en minúsculas para todos los tuits del conjunto de datos. De esta manera, se cuenta con un número menor de reglas de conversión y combinaciones. Esto debido, a que de cara al análisis de caracteres, no es lo mismo encontrar una letra mayúscula a una escrita en minúscula y requeriría un mayor tiempo de procesamiento para cada palabra. También se incluye la eliminación de caracteres no deseados y únicamente fueron permitidos los que se registraron como parámetro en la función para su extracción.

Posteriormente, el resultado de la transformación pasó por el componente encargado de extraer cada una de las palabras como elementos independientes en el contenido del

tuit. El carácter de espacio en blanco es el utilizado como el separador estándar. De este proceso, fue obtenida una estructura de datos de tipo lista donde cada una de las palabras es asociada como un elemento individual para su procesamiento.

Las listas resultantes pasaron luego por otro componente. Este fue el encargado de la aplicación de las reglas de conversión de caracteres que fueron definidas para el proceso. Se establecieron las mismas reglas utilizadas en el proceso de conversión a formato *leet* para todas las vocales y se adicionó una nueva regla donde el carácter "@" también puede corresponder a una letra "a". Este componente analizó carácter a carácter la totalidad de las palabras verificando si cumplían con alguna de las normas establecidas. Si no hubo coincidencia, el carácter resultante fue el mismo que se validó. Si había coincidencia con una regla, el carácter fue reemplazado por el indicado por los parámetros establecidos. Y finalmente, si el carácter contaba con dos o más reglas válidas, se realizó la extracción de todos los caracteres equivalentes. Como objeto para el almacenamiento de los resultados, se definió una nueva estructura de datos del tipo lista para las palabras individualmente, donde cada uno de los elementos que la constituyen corresponden al carácter o caracteres obtenidos con la validación de las reglas. Las listas resultantes de este proceso se anidaron en una lista general.

La generación de palabras constituyó la siguiente etapa del proceso. En esta, se tomaron separadamente las letras en cada lista anidada y se derivaron todas las posibles combinaciones de palabras de acuerdo con el número de opciones de letra obtenidos. Si de esta combinación se generó una única palabra, se tomó como el resultado final. Si se generó más de una, cada una de ellas se valida contra un diccionario de idioma inglés y únicamente se tomaron como válidas aquellas que se encontraron en el diccionario. Finalmente, si no se hallaron coincidencias en el diccionario, todas las palabras generadas se seleccionaron como válidas (asociadas a posibles errores de ortografía o caracteres intercambiados) y fue responsabilidad del siguiente módulo de hacerse cargo de este último escenario. Como resultado de este proceso, se obtuvo una nueva lista donde cada uno de los elementos corresponde a cada palabra, y si fue obtenida más de una, se define una lista anidada con el total de palabras resultantes.

Por último, se realizó la construcción de todas las posibles frases resultantes utilizando la lista obtenida. Si la lista únicamente contaba con elementos de tipo no lista, se produjo una única frase. En caso de haber más de una palabra válida, se formularon todas las frases posibles de acuerdo con el número de palabras posibles a combinar. Como resultado

del módulo, se generó un archivo separado por comas con dos elementos; el primero, compuesto por una lista con la frase o frases resultantes; y el segundo corresponde a la etiqueta de clasificación de tendencia para el tuit procesado. El archivo formado es la interfaz de entrada al módulo de calificación.

El diseño para el módulo de conversión con un tuit escrito en formato *leet*, se representa de manera gráfica en la figura 4.6.

4.3.2 Módulo de calificación

Con la interfaz de entrada construida en el módulo de conversión, se inició con la definición del proceso que efectua la calificación de las diferentes frases obtenidas. De esta manera, se derivaron las mejores opciones con respecto al análisis realizado por los modelos de lenguaje utilizados. En primer lugar, se eligieron todas las frases en el archivo de entrada y se cargaron en una estructura de datos para facilitar su manipulación. Posteriormente, se realizó el cargue de los modelos de lenguaje preentrenados. Se fijó un proceso responsable de enviar las oraciones a cada uno de los tres modelos extraer de este modo una calificación según el modelo consultado. Estos resultaron se cargaron en una estructura de datos y se seleccionó la frase con la mejor calificación para cada modelo. Finalmente, crearon tres archivos separados por comas con dos elementos; el primero compuesto por la frase final extraida de la conversión del tuit en formato *leet* a texto claro, y el segundo contiene la etiqueta de clasificación de tendencia para el tuit procesado. Este archivo generado es la interfaz de entrada al módulo de evaluación.

El diseño para el módulo de calificación utilizando el mismo tuit de ejemplo en la fase de diseño, se representa en la figura 4.7.

4.4 Fase de desarrollo

Con los diseños resultantes, se desarrolló la codificación de cada uno de ellos utilizando como lenguaje de programación Python 3.7 mediante *Google Colaboratory*. Una de las principales razones para utilizar esta herramienta, es la facilidad de acceder a módulos GPU de alta velocidad debido a que las tareas de validación de puntaje contra los módulos de lenguaje demandan una alta capacidad de recursos computaciones. Una CPU estándar tardaría un tiempo considerable en ejecutar este tipo de tareas, es por ello, que las GPU permitieron que los procesos fueran más rápidos en su ejecución. Como consecuencia en

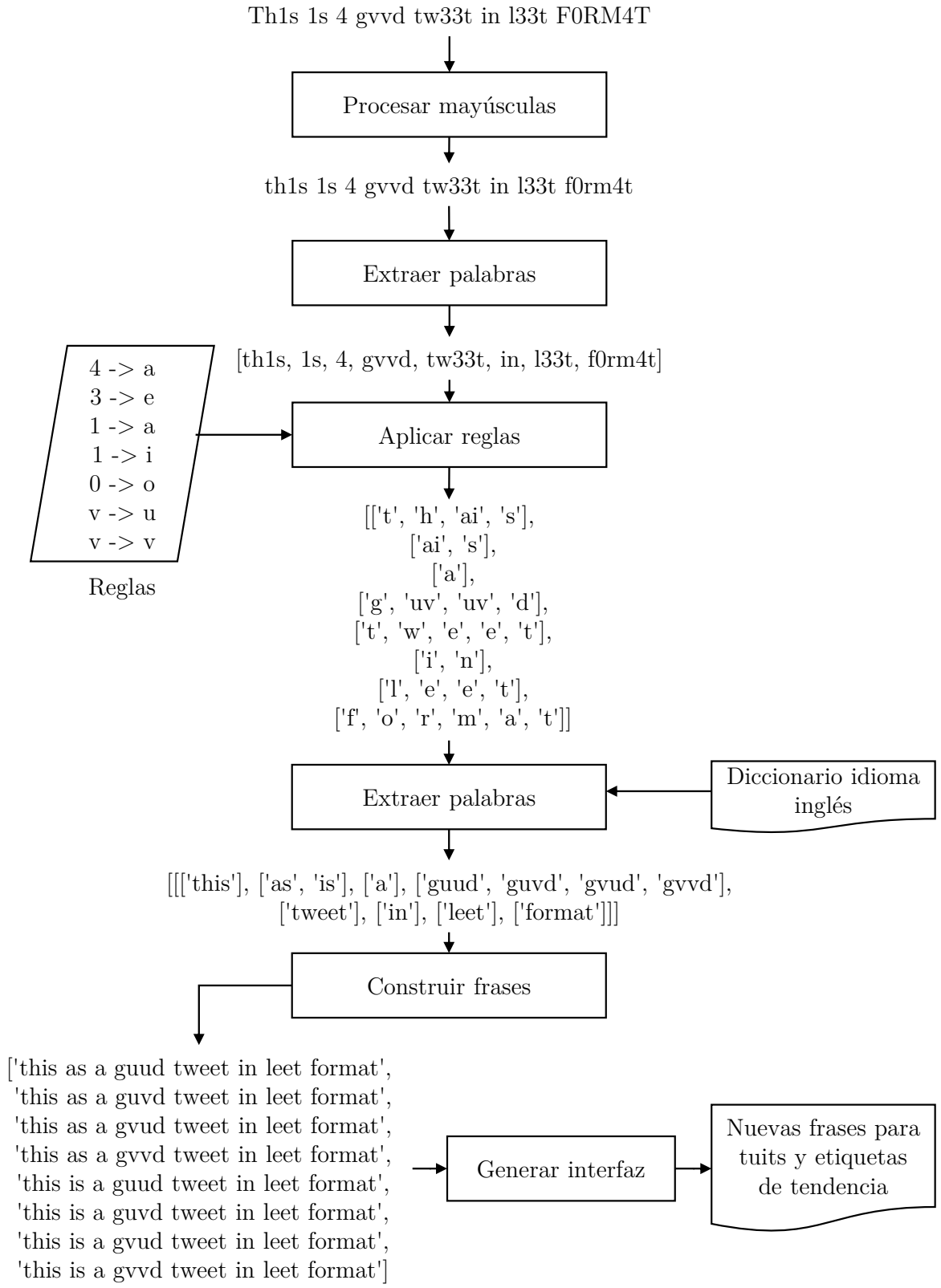
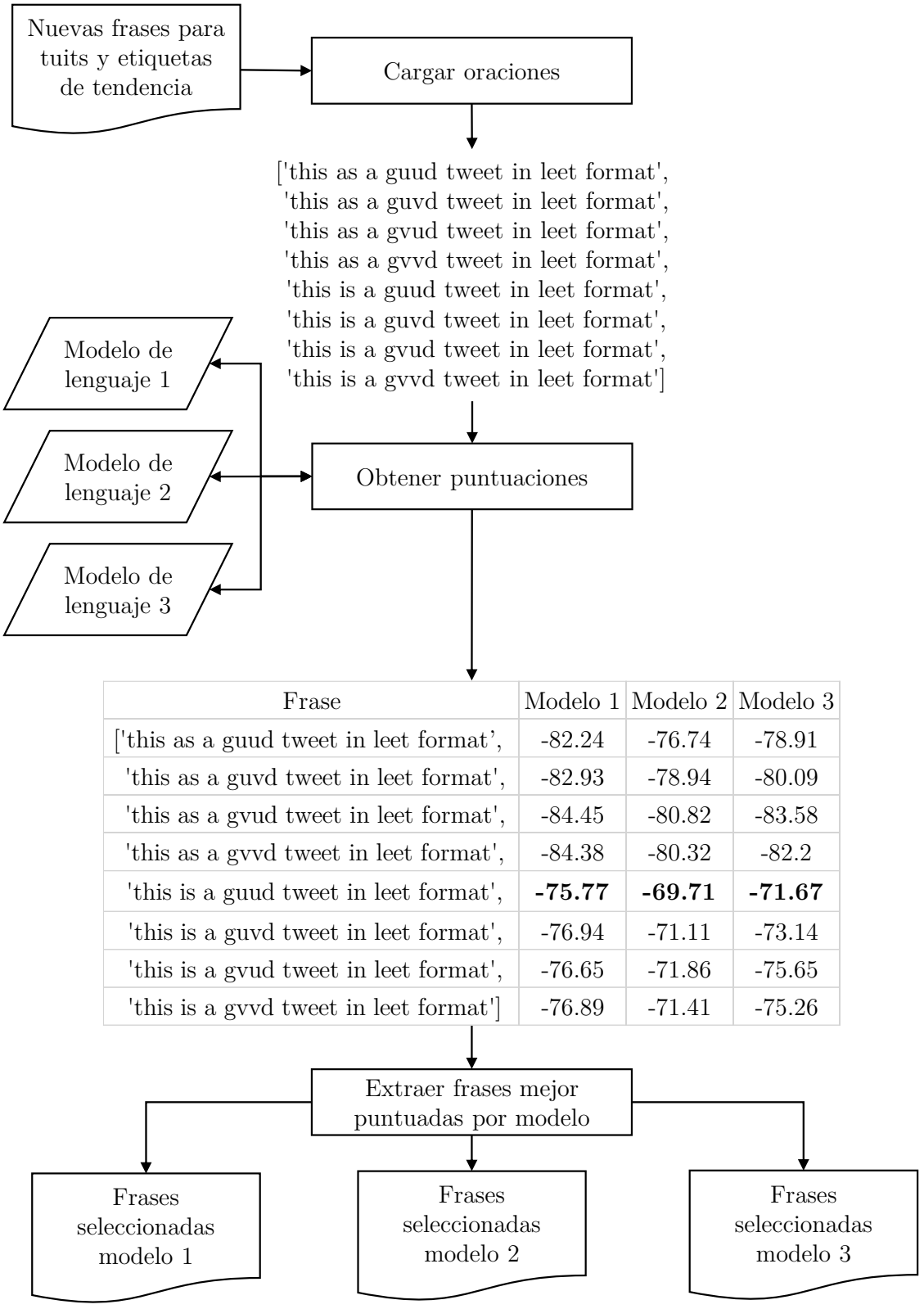


Figura 4.6: Diseño final para la fase de conversión



Frase	Modelo 1	Modelo 2	Modelo 3
['this as a guud tweet in leet format',	-82.24	-76.74	-78.91
'this as a guvd tweet in leet format',	-82.93	-78.94	-80.09
'this as a gvud tweet in leet format',	-84.45	-80.82	-83.58
'this as a gvvd tweet in leet format',	-84.38	-80.32	-82.2
'this is a guud tweet in leet format',	**-75.77**	**-69.71**	**-71.67**
'this is a guvd tweet in leet format',	-76.94	-71.11	-73.14
'this is a gvud tweet in leet format',	-76.65	-71.86	-75.65
'this is a gvvd tweet in leet format']	-76.89	-71.41	-75.26

Figura 4.7: Diseño final para la fase de calificación

la fase de desarrollo, se presentan los componentes implementados para los módulos de conversión y los de clasificación.

4.4.1 Implementación módulo de conversión

La primera implementación consistió en convertir todos los caracteres de mayúsculas a minúsculas. Seguido por la eliminación de todos los caracteres inválidos en los tuits, es decir, todos aquellos caracteres diferentes a los listados en la función. Y finalmente, se retornaron todas las palabras como elementos independientes en una lista. El segmento de componente implementado para esta primera función se presenta en el código 1.

```
1    def ExtraerPalabras(Cadena):
2        Cadena = Cadena.lower()
3        CadenaSinCaracteres = \
4          re.sub(r"[^a-zA-Z0-9\n áéíóúñü%@!\"#$%&/()=?â¡\'â¿â´+{[^}`]-.|;']", \
5          "",Cadena)
6        return CadenaSinCaracteres.split()
```

Código 1: Código de función ExtraerPalabras

Posteriormente, se realizó la implementación de las reglas de conversión para cada una de las letras de acuerdo con las transformaciones concebidas mediante el uso de la estructura de datos diccionario. El detalle de las reglas codificadas se presenta en el código 2.

```
1        reglas_conversion_leet = {
2            'a' : ['4','@'],
3            'e' : '3',
4            'i' : '1',
5            'o' : '0',
6            'u' : 'v',
7            'v' : 'v'
8        }
```

Código 2: Reglas de conversión de caracteres *leet* a su letra correspondiente

La función encargada aplicar las reglas de codificación *leet* a su correspondiente carácter o caracteres, se denominó *aplicar_reglas* obteniendo así como retorno, un conjunto de listas anidadas donde cada una corresponde a una palabra, y los elementos que componen cada una de estas sublistas es la letra o letras obtenidas luego de aplicadas las reglas. Si un

carácter no es encontrado en las reglas, pasa directamente desde su posición en la palabra original. Si un carácter es encontrado una o más veces en las reglas, el o los caracteres equivalentes fueron los utilizados para cargar los elementos de la lista. El código 3 presenta la implementación de esta función.

```
1         def Aplicar_Reglas (Tokens):
2
3           Total_Letras = []
4
5           for Palabra in Tokens:
6
7             lista = []
8
9             for Letra in Palabra:
10              list_of_keys = [key
11                              for key, list_of_values in reglas_conversion_leet.items()
12                              if Letra in list_of_values]
13
14              if list_of_keys:
15                  if len(list_of_keys) > 1:
16                    lista.append(''.join(list_of_keys))
17                  else:
18                    lista = lista + list_of_keys
19              else:
20                  lista = lista + list(Letra)
21
22            Total_Letras.append(lista)
23
24          return Total_Letras
```

Código 3: Código de función Aplicar_Reglas

El paso siguiente consistió en tomar todos los caracteres posibles obtenidos y realizar el ensamblaje de cada uno de ellos en la palabra o todas las posibles que pudiesen generarse con base en la combinación de los caracteres extraídos. Luego, si alguna combinación de letras produjo más de una como opción, fueron sometidas a un procedimiento de validación con un diccionario en idioma inglés. Para dicha validación se utilizó la librería *enchant* que posee un amplio vocabulario del idioma inglés, funcionando de manera conveniente para esta tarea. Las palabras encontradas en el diccionario fueron aceptadas como válidas. Si del conjunto de palabras ninguna fue confirmada por el diccionario (ya sea por mala escritura o por errores ortográficos), todo el conjunto fue seleccionado como válido para continuar en el proceso. Finalmente, la función entregó un consolidado de tipo lista con

sublistas anidadas, donde cada una de estas contiene un único elemento, si de la combinación se generó una única palabra. Si se generaron dos o más, se crean como elementos en su sublista correspondiente. El código 4 presenta la implementación de esta función.

```
def Armar_Palabras (Lista_Palabras):
  tamaño = 0
  Palabras_Unidas = []

  for letras in Lista_Palabras:
    Palabras = [''.join(word) for word in product(*letras)]
    tamaño = len(Palabras)

    if tamaño > 1:
      min = 0
      Palabras_Borrar = []

      for palabra in Palabras:
        Val = Diccionario_Ingles.check(palabra)

        if not Val:
          Palabras_Borrar.append(palabra)
          min = min + 1
      l3 = [x for x in Palabras if x not in Palabras_Borrar]
      Palabras = l3

      if min == tamaño:
        Palabras.append(palabra)
    Palabras_Unidas.append(Palabras)

  return Palabras_Unidas
```

Código 4: Código de función Armar_Palabras

Finalmente, para realizar el armado de las frases finales, se crearon dos funciones para ejecutar esta construcción. La primera de ellas toma todas las palabras resultantes de la función anterior, y realiza su combinación. Esta función generó una lista con conjuntos en su interior. Cada una de las agrupaciones corresponde a la frase o frases que se produjeron al combinar las palabras. Y la segunda función tomó cada uno de los conjuntos extraídos para así finalmente crear una lista con elementos de tipo cadena de texto, donde la totalidad de los elementos de la lista ya corresponde a cada una de las opciones que va a reemplazar al tuit original. El detalle de las funciones se presenta en el código 5.

En ultima instancia, las oraciones generadas se guardaron en un archivo formato CSV con dos columnas. La primera denominada Frase_Final, que cuenta con todas las frases

```
1     def Armar_Frases (lista):
2       Frases_Finales = []
3       Frases = list(it.product(*lista))
4       return Frases
5
6     def Frase_Final (lista):
7      Frases = []
8      for item in lista:
9        Frases.append(' '.join(item))
10     return Frases
```

Código 5: Código de funciones Armar_Frases y Frase_Final

candidatas a sustituir al tuit original. Y la segunda, denominada *Suicide*, que corresponde a un "1" si el tuit fue clasificado con tendencia suicida, o con "0" en caso contrario. Este archivo de salida fue el insumo de entrada para el módulo de calificación.

4.4.2 Implementación módulo de calificación

La implementación de este módulo fue ejecutada utilizando como eje de funcionamiento la interfaz *lm-scorer*, presenta facilidad en su uso para poder acceder a diferentes módulos de lenguaje preentrenados. De manera que, al realizarse algunas modificaciones en la forma de ejecutarse, fue posible aplicar el proceso de evaluación para todo el contenido del archivo CSV recibido del módulo de conversión.

De acuerdo con el diseño, se seleccionaron los tres modelos de lenguaje *DistilGPT2*, *GPT2* y *GPT2-Medium* para continuar la implementación de esta etapa.

Cada una de las frases procesadas por tuit, se almacenaron en un archivo temporal el cual se reemplazaba en cada una de las iteraciones y luego se enviaba como parámetro en la instrucción para el cálculo de puntiación con el modelo de lenguaje. Si únicamente se obtuvo una frase a analizar para alguno de los tuits, no se envió al método para obtener su puntuación, sino que se asignó directamente el valor "1". En caso contrario, se obtienen todos los puntajes para cada una de las frases. La función para obtener el puntaje se representa en el código 6.

Como consecuencia, se obtuvo la frase final tomando todas las puntuaciones resultantes de los modelos de lenguaje utilizados, se buscó la mayor de ellas y se extrajo la frase correspondiente a dicha puntuación. De esta manera, se obtuvo el tuit final que sustituyó al tuit escrito en formato *leet*. La función para la extracción del tuit final se representa en

```
def Obtener_Puntaje (lista):
  os.remove("/content/drive/MyDrive/Datos/Suicidio/sentences.txt")
  f = open("/content/drive/MyDrive/Datos/Suicidio/sentences.txt", "a")
  Score = []
  if len(lista) > 1:
    for item in lista:
      f.write(item)
      f.write("\n")
    f.close()
    RE = !lm-scorer -m $MODEL -r $REDUCE -b $BATCH_SIZE --cuda $CUDA -lp \
    /content/drive/MyDrive/Datos/Suicidio/sentences3.txt
    for item in RE:
      Score.append(item)
  else:
      Score.append("1")
  return Score
```

Código 6: Código de función Obtener_Puntaje

el código 7.

```
def Obtener_Tweet_Final (Frase, Score):
  Resultado = ""
  Cadenas = []
  Puntajes = []

  for calificacion in Score:
    if calificacion == '1':
      Resultado = Frase[0]
    else:
      split_string = calificacion.split("\t")
      Cadenas.append(split_string[0])
      Puntajes.append(float(split_string[1]))

  if Resultado == "":
    max_ind = Puntajes.index(max(Puntajes))
    Resultado = Cadenas[max_ind]
  return Resultado
```

Código 7: Código de función Obtener_Tweet_Final

4.5 Fase de Prueba

Al ejecutarse los componentes implementados en la fase de desarrollo con los tuits en formato *leet*, se obtuvo un archivo en formato CSV con dos columnas. La primera llamada

Tweet, que contiene el listado de todos los tuits modificados, y la segunda denominada *Suicide* que corresponde a un "1" si el tuit fue clasificado con tendencia suicida, o con "0" en caso contrario. Este archivo fue el insumo de entrada para el proceso de pruebas y para confirmar si las transformaciones efectuadas cumplen con el objetivo de clasificación.

Para probar el correcto funcionamiento del modelo implementado, como primer paso, se realizó la validación con una muestra del conjunto total de datos; 20 tuits en total, 10 de ellos con etiqueta de tendencia suicida y 10 de ellos no. Los tuits utilizados y su clasificación se presentan en el cuadro 4.3. Posteriormente, se ingresaron los 20 tuits al modelo previamente entrenado para verificar la asignación de etiqueta con el modelo entrenado para cada uno de los trinos en el proceso de evaluación para verificar su predicción inicial.

Como siguiente evento de prueba, se realizó la modificación *leet* para los primeros ocho tuits clasificados con tendencias suicidas, y a dos de ellos no se realizó ningún tipo de cambio; de esta manera, se tienen como datos de control para la nueva prueba de clasificación. El procedimiento se realizó con nuevas reglas de codificación para algunos de los caracteres del tuit, es decir, que en algunos de ellos la "a" fue reemplazada por un "@" y en otros fue reemplazada por un "4". Las reglas de las letras modificadas se presentan en el cuadro 4.4. Y los tuits finales con las modificaciones efectuadas, se presentan en el cuadro 4.5.

Nuevamente, se ejecutó el proceso de clasificación con los 20 tuits, incluyendo los 8 que han sido modificados con la codificación *leet*. Así se confirmaría que efectivamente el resultado de la clasificación ha sido afectado por el cambio en el contenido de cada uno de los tuits.

Al comprobar que los resultados fueron satisfactorios, se procedió realizar la prueba definitiva del modelo implementado. Para ello, se tomó el archivo que generó el peor desempeño en su clasificación en la fase de comprobación. Este archivo fue en el que se realizó la modificación de todas las vocales por un caractér *leet* equivalente. Se procesó y se generó un archivo transformado con las frases equivalentes. Este archivo fue ingresado al modelo entrenado en la fase inicial. Los resultados de los experimentos serán presentados en la siguiente sección.

Tabla 4.3: Tuits y su clasificación

Tuit	¿Tendencias suicidas?
One day ill kill myself and my suicide note will just be some shit like “my arms are kinda chubby”. Bye”	Sí
Thinking in suicide has now become my new world and I am desperately searching for answers	Sí
Today, one year ago, I lost a neighbour, and friend, to. So here is a quote that got me through my worst times.	Sí
I need a guidance, mentor, father figure, guardian angel or whatever fuck out there that can tell me what to do with me	Sí
Im a fucking lunatic when I get angry its so terrible I need to work on this so bad before I end up doing somethig against me	Sí
Just wanna fucking sleep and never wake up ever again	Sí
I fucking miss you but youre probably better off without me.	Sí
me, when I finally die after saying I want to die every single day	Sí
Another year has passed and I am still alone, sad and suffering. Will I ever live with dignity? Or will not?	Sí
Im tired of lies and living in lies, of people showing only the versions they want to show and of shows that people put on.	Sí
Delighted to join the Centenary Celebrations of Patna University. Sharing my speech during the programme today.	No
Focus on opportunities more than you focus on money.	No
If you find someone that makes you happy, enjoy it... life is not fair nor give many opportunities...	No
Just keep going forward, no matter what	No
Jealousy is important for a person to move ahead in life. It keeps you on your toes and helps you realize	No
Completely devestated by the fires affecting my community. I am one of the lucky ones. My animals and LOVE OF MY LIFE made it easier	No
normal workday for me. I do still have plans/aspirations to submit some info for the miz mag - honest i do.	No
Watching This Morning and deciding what to do with myself today. Think I’ll keep searching for that elusive job!	No
It snowed! Okay, there’s not enough to build Frosty, but it snowed. As long as it’s cold, it might as well snow...Enjoy your day	No
Hurrah! Someone is listening! Delay so long, have now missed meeting. Back home to walk dog and reschedule	No

Tabla 4.4: Reglas de codificación para la fase 2

Letra Original	Letra modificada
a	@, 4
l	1
i	1
o	0
u	v

Tabla 4.5: Tuits luego de aplicar modificación *leet*

Tuit	¿Tendencias suicidas?
0n3 d@y 111 k111 mys3lf @nd my sv1c1d3 n0t3 w111 jvst b3 s0m3 sh1t 11k3 my @rms @r3 k1nd@ chvbby By3	Sí
Th1nk1ng 1n Sv1c1d3 h@s n0w b3c0m3 my n3w w0rld @nd 1 @m d3sp3r@t3ly s3@rch1ng f0r @nsw3rs	Sí
T0d@y 0n3 y3@r @g0 1 l0st @ n31ghb0vr @nd fr13nd t00 S0 h3r3 1s @ qv0t3 th@t g0t m3 thr0vgh my w0rst t1m3s	Sí
1 n33d 4 gv1d4nc3 m3nt0r f4th3r f1gvr3 gv4rd14n 4ng3l 0r wh4t3v3r fvck 0vt th3r3 th4t c4n t3ll m3 wh4t t0 d0 w1th m3	Sí
1m 4 fvck1ng lvn4t1c wh3n 1 g3t 4ngry 1ts s0 t3rr1bl3 1 n33d t0 w0rk 0n th1s s0 b4d b3f0r3 1 3nd vp d01ng s0m3th1g 4g41nst m3	Sí
Jvst w4nn4 fvck1ng sl33p 4nd n3v3r w4k3 vp 3v3r 4g41n	Sí
1 fvck1ng m1ss y0v bvt y0vr3 pr0b4bly b3tt3r 0ff w1th0vt m3	Sí
m3 wh3n 1 f1n4lly d13 4ft3r s4y1ng 1 w4nt t0 d13 3v3ry s1ngl3 d4y	Sí
Another year has passed and I am still alone sad and suffering Will I ever live with dignity? Or will not?	Sí
Im tired of lies and living in lies of people showing only the versions they want to show and of shows that people put on	Sí

Parte IV

Resultados experimentales y discusión

Capítulo 5

Resultados de experimentos

5.1 Resultados fase de entrenamiento

Como resultado de la fase de entrenamiento del modelo entrenado inicial, generó como métricas de desempeño, un *Accuracy* del 95,58%, un valor F1 de 95,43%, *Precision* del 92,42% y *Recall* del 98,64%. En la figura 5.1 se presenta la curva ROC obtenida en la fase de entrenamiento; al igual que en la figura 5.2 se presentan los resultados de clasificación iniciales.

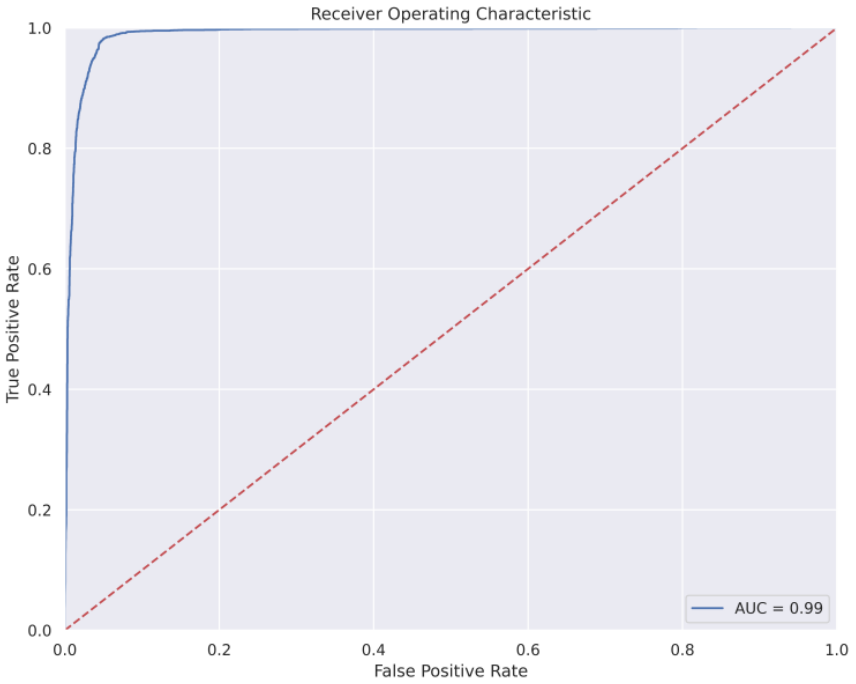


Figura 5.1: Curva ROC y ROC AUC para la fase de entrenamiento del modelo.

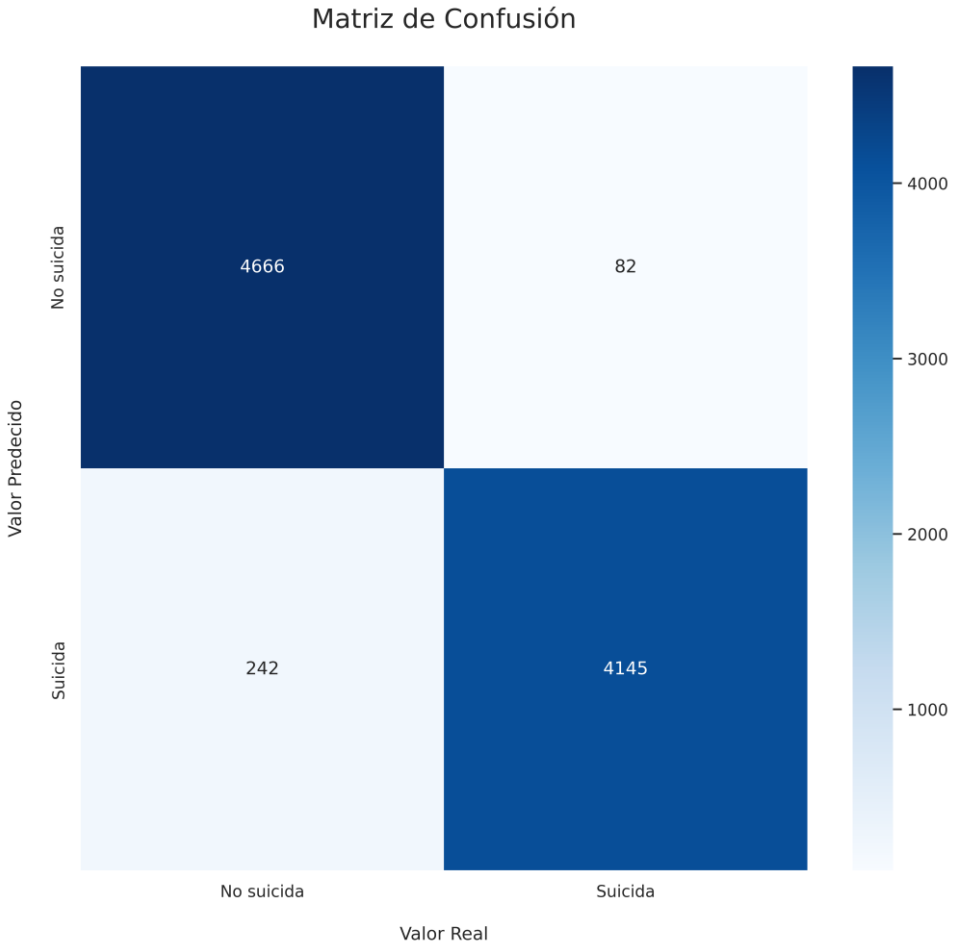


Figura 5.2: Matríz de confusión para la fase de entrenamiento del modelo.

Debido a que la modificación de datos fue aplicada únicamente sobre los tuits etiquetados con contenido de tendencias suicidas, aquellos que tienen la etiqueta no suicida (clasificados 4666 correctamente y 242 como falsos negativos), no serán tenidos en cuenta para las demás fases experimentales. Esto debido que para esta categoría, se obtuvo el mismo resultado en todos los experimentos.

5.2 Resultados fase de comprobación

5.2.1 Modificación de palabras

La tabla de los resultados obtenidos para la etapa se plasma en el cuadro 5.1, y la comparación entre las curvas ROC entre el conjunto de datos sin modificar y el último de los escenarios, se presenta en la figura 5.3.

En cada uno de los experimentos llevados a cabo, se comprobó que al aumentar el

Tabla 5.1: Métricas de evaluación y resultados para clasificación de tuits con modificación de palabras clave

Experimento	*Accuracy*	*F1 score*	*Precision*	*Recall*	*ROC - AUC*	Tuits suicidas	Tuits no suicidas
Fase de entrenamiento	96.450%	96.240%	94.480%	98.060%	0.989	82	4145
Grupo de palabras 1	96.170%	95.920%	94.450%	97.440%	0.987	108	4119
Grupo de palabras 1 y 2	94.750%	94.320%	94.280%	94.370%	0.983	238	3989
Grupo de palabras 1,2 y 3	94.010%	93.480%	94.190%	92.780%	0.981	305	3922
Grupo de palabras 1,2,3 y 4	92.810%	92.070%	94.030%	90.180%	0.978	415	3812

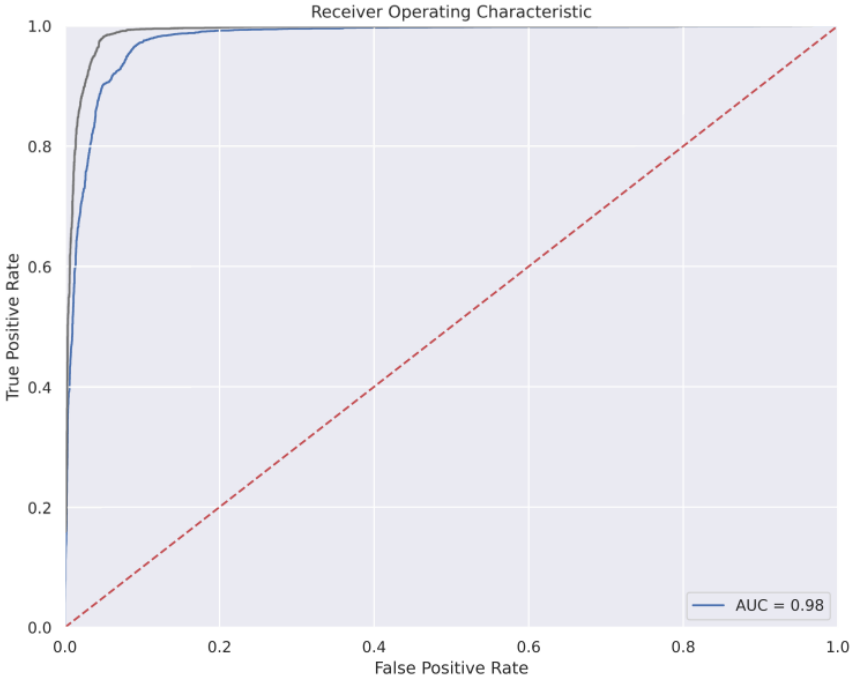


Figura 5.3: Comparación ROC entre el experimento con los datos originales (línea gris) y el experimento con la modificación de las 20 palabras seleccionadas (línea azul)

número de palabras utilizadas en *leet speaking*, el desempeño del modelo se ve afectado negativamente. Esto se observa en la figura 5.4. Confirmando así, que existe una correlación negativa; debido a que al aumentar el número de palabras utilizadas en *leet speaking*, disminuye la efectividad de clasificación del modelo.

A pesar que las evaluaciones de desempeño del modelo al finalizar los experimentos se encuentran sobre el 90%, es válido deducir preliminarmente, que el modelo entrenado sigue siendo bueno matemáticamente hablando en la tarea de clasificación. Pero, al evaluar la tendencia, entre más palabras sean escritas en forma *leet speaking*, aumenta el número de tuits no detectados con tendencias suicidas que pueden ser reales, pasándolos por alto y evitar así tomar las acciones a seguir sobre estos usuarios. En un clasificador de este tipo, una sola predicción errónea puede ser la diferencia entre identificar a una persona con este tipo de tendencia para así dar un seguimiento, tratarla adecuadamente y brindar la respectiva orientación, o por el contrario, omitir algunos casos y que puedan llegar a un desenlace desafortunado.

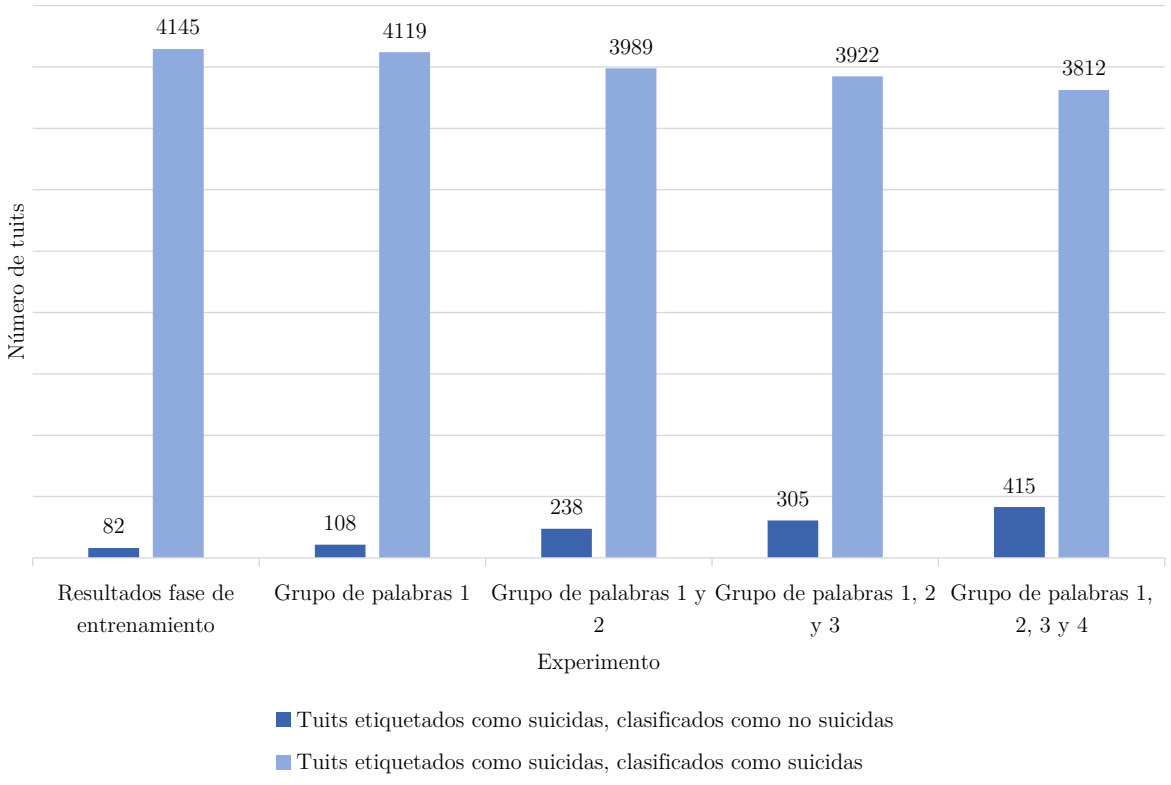


Figura 5.4: Resultados de clasificación modificando palabras clave en los tuits

5.2.2 Modificación de vocales

La tabla de los resultados obtenidos para la etapa se plasma en el cuadro 5.2. La comparación entre las curvas ROC entre el conjunto de datos sin modificar y el escenario donde todas las vocales fueron modificadas, se presenta en la figura 5.5.

De acuerdo con lo concluido por [43], en inglés, la frecuencia de las vocales en difer-

Tabla 5.2: Métricas de evaluación y resultados para clasificación de tuits con modificación de vocales

Experimento	*Accuracy*	*F1 score*	*Precision*	*Recall*	*ROC - AUC*	Tuits suicidas	Tuits no suicidas
Fase de entrenamiento	96.450%	96.240%	94.480%	98.060%	0.989	82	4145
“a” por “4”	83.850%	80.240%	92.520%	70.830%	0.924	1233	2994
“e” por “3”	79.170%	72.950%	91.380%	60.700%	0.915	1661	2566
“i” por “1”	89.140%	87.520%	93.490%	82.260%	0.951	750	3477
“o” por “0”	80.100%	87.460%	93.490%	82.160%	0.954	754	3473
“u” por “v”	84.320%	80.920%	92.620%	71.850%	0.905	1190	3037
Todas las vocales	51.990%	3.650%	25.540%	1.960%	0.806	4144	83

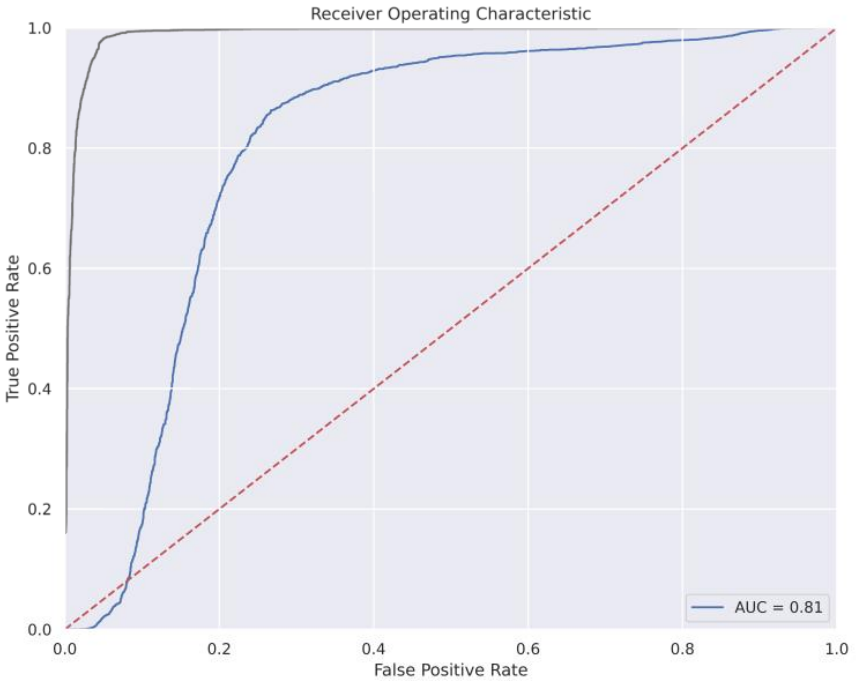


Figura 5.5: Comparación ROC entre el experimento con los datos originales (línea gris) y el experimento con la modificación de todas las vocales (línea azul)

entes fuentes textuales se identifica en orden descendente con la siguiente distribución: E (11,8%), A (7,7%), I (7,3%), O (7,2%) y finalmente la U (2,6%). De acuerdo con los resultados del desempeño para cada uno de los experimentos que se ejecutaron en esta etapa, presentados en el cuadro 4.2, es posible observar un comportamiento bastante interesante. La disminución del desempeño asociado a la vocal modificada, ordenado del peor al mejor desempeño, se evidencia en el siguiente orden: E, A, U, O, I. En este comportamiento, es posible identificar que la U a pesar de que su frecuencia en textos es aproximadamente un

4,65% inferior a la I y la O, genera un impacto mayor en el desempeño. Esta situación podría dar un alcance a un estudio posterior, para así analizar e identificar alguna relación en que la vocal U se encuentra asociadas a palabras que incrementan la detección de tendencias suicidas en tuits y prestar mayor atención a estas palabras.

Finalmente, el experimento en el que se realizó la modificación a todas las vocales, se puede evidenciar que las medidas de desempeño del modelo son bastante negativas. Ocasionando así que se pase por alto cualquier detección de tendencias suicidas con el clasificador entrenado, y a pesar que su desempeño con el conjunto de datos de prueba es muy bueno, este tipo de modificaciones ocasionan que la tarea de clasificación no sea precisa y que el modelo resulte obsoleto para esta tarea en particular. En la matriz de confusión presentada en la figura 5.2, es posible observar que el resultado en el modelo con los datos originales, únicamente 82 casos de tuits con tendencias suicidas fueron etiquetados como no suicidas. Por otro lado, 5.6 se confirma que 4144 tuits con tendencias suicidas fueron clasificados con la etiqueta contraria y únicamente 83 fueron clasificados de manera correcta.

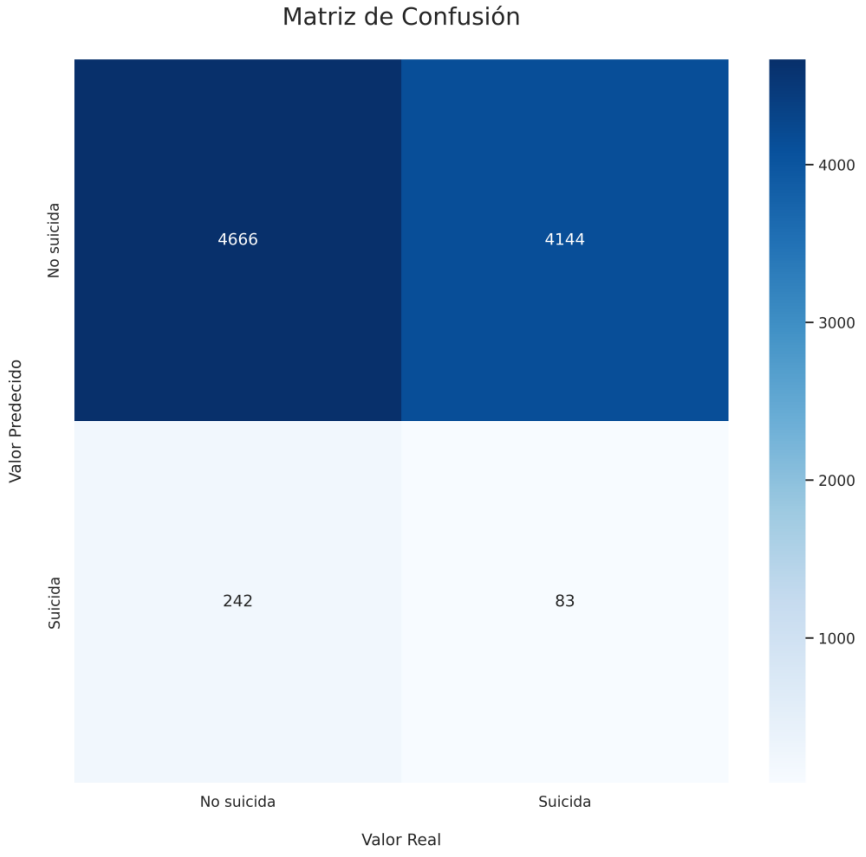


Figura 5.6: Matriz de confusión para el conjunto de datos con todas las vocales modificadas

Por último, el desempeño del modelo para los experimentos donde se realizó la modificación en grupos de vocales se presenta en la figura 5.7.

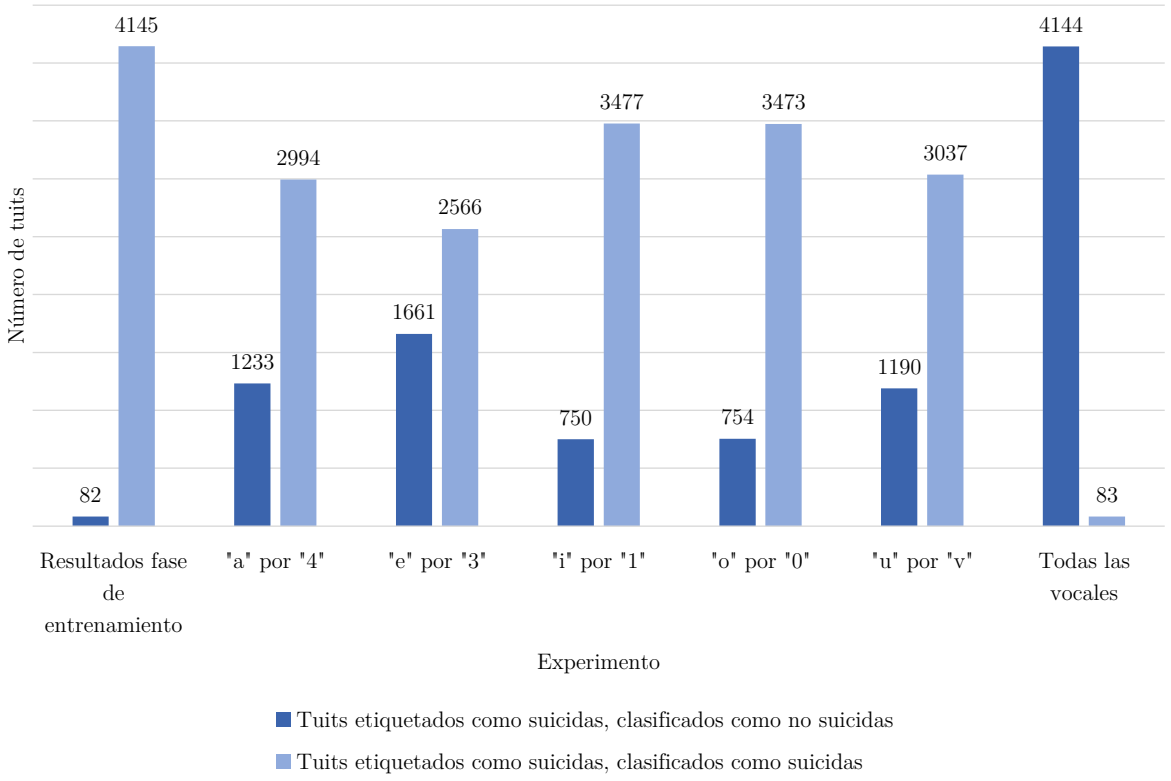


Figura 5.7: Resultados de clasificación de los experimentos de la etapa 2

5.3 Resultados fase de diseño, implementación y pruebas

El resultado de esta clasificación es el ideal, ya que cada uno de los tuits fue clasificado de manera correcta con respecto a su etiqueta. Esto se puede confirmar en la sección izquierda de la figura 5.8.

En la figura 5.8 se confirma que únicamente uno de los tuits modificados ha sido clasificado con la etiqueta de suicida además de los dos tuits a los que no se les realizó ningún tipo de modificación. Los restantes siete fueron clasificados como no suicidas, a pesar que previamente comprobamos que sí lo son.

Al haberse comprobado nuevamente los resultados de la fase de comprobación, se procedió a comprobar el módulo *leet* implementado. Tomando el archivo total con los 20 tuits que se ingresaron al clasificador, y aplicando el nuevo procesamiento implementado. Esto con el objetivo de convertir cada uno de los tuits a un formato sin codificación *leet*,

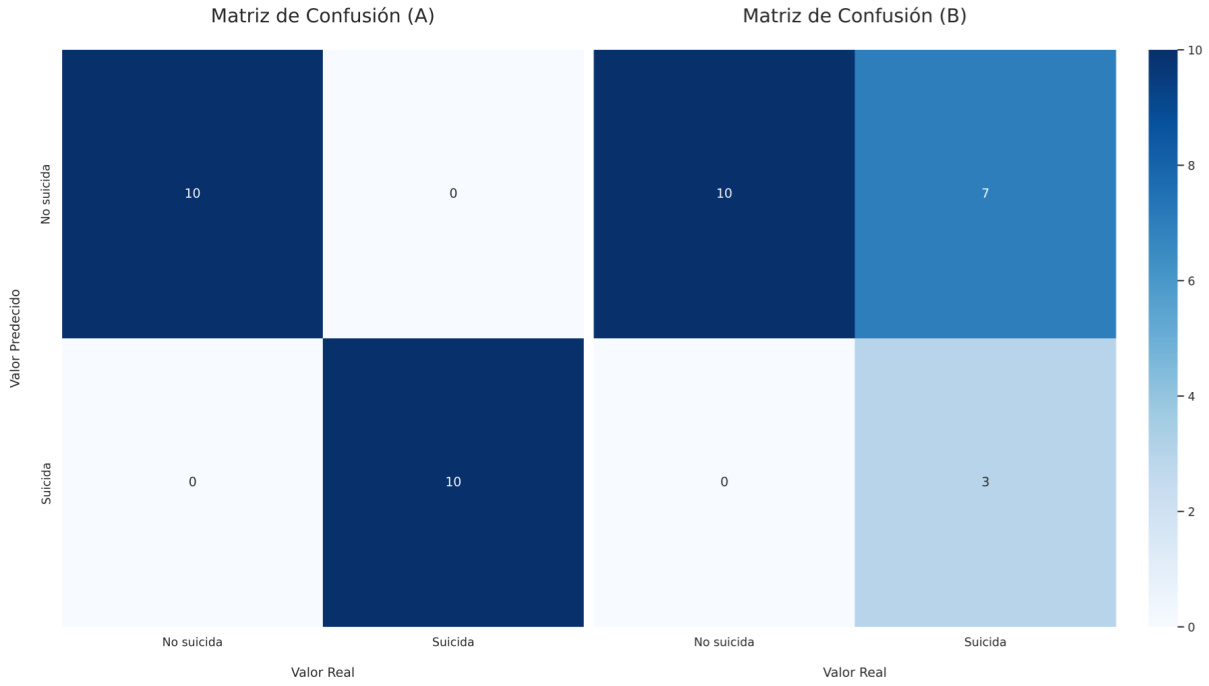


Figura 5.8: Matriz de confusión para los datos de prueba originales (A) y datos modificados (B)

para que al momento de volver a ingresarse al modelo de clasificación, genere el resultado de clasificación correcto.

Debido a que en los tuits tenemos casos donde un mismo carácter *leet* puede estar asociado a dos letras diferentes (el 1 puede ser L o I), o también un carácter puede estar asociado así mismo como a otro (la V puede ser V o U), es necesario que todas estas combinaciones sean tenidas en cuenta al momento de querer llegar al mensaje original.

El siguiente paso consistió en verificar cuál de las opciones generadas es la mejor de acuerdo a las palabras y estructura gramatical de las mismas. Para ello, se toma el resultado generado por el componente desarrollado *leet* y se ingresa al modelo obtenido en la fase de entrenamiento para que así, se efectúe la calificación de todas las posibles opciones, y genere así una puntuación con base en el análisis de cada una de las oraciones. Finalmente, la oración con la puntuación más alta será la seleccionada; y de esta manera se obtiene como resultado una única homologación al tuit original en formato *leet*.

Por ejemplo, para el tuit:

T0d@y 0n3 y3@r @g0 1 l0st @ n31ghb0vr @nd fr13nd t00 S0 h3r3 1s @ qv0t3 th@t g0t

m3 thr0vgh my w0rst t1m3s

Se generaron un total de 16 frases que debieron ser analizadas como resultado de aplicar el convertidor *leet*. Cada una de estas frases fueron analizadas por los tres modelos de lenguaje y se les asignó calificaciones a cada una de ellas. Las frases resultantes y su correspondiente puntuación para cada uno de los modelos, se presentan en el cuadro 5.3. Se resalta en negrillas la opción con la mejor puntuación en cada modelo, que corresponde a la frase seleccionada como opción final que sustituirá al tuit en formato *leet*.

Fueron ingresados nuevamente al clasificador los resultados obtenidos del módulo de conversión para verificar si el desempeño es el mismo a la ejecución con el conjunto de datos sin ningún tipo de modificación *leet*. Como resultado, se obtuvo de nuevo la matriz de confusión (A) de la imágen 5.8. Esto permite deducir que el resultado es el deseado y que los tuits asociados a tendencias suicidas fueron clasificados correctamente, y que la transformación no afectó a aquellos tuits que no cuentan con este tipo de tendencia, por lo que para este conjunto de datos, puede concluirse que el resultado de la prueba es satisfactorio.

Como resumen final de esta fase, se presenta el cuadro 5.4 con el objetivo de comparar el tuit que ingresó al procesador *leet*, cuantas oraciones se generaron por cada tuit, la puntuación obtenida por cada uno de los modelos de lenguaje, el tuit obtenido al final del proceso y si el resultado seleccionado es igual o no al tuit original previo a la codificación *leet*.

Tabla 5.3: Resultado de módulo *leet* y de calificación de frases

Oración	Distil GPT2	GPT2	GPT2 Medium
today one year ago i lost a neighbour and friend too so here is a quote that got me through my worst times	**-114.6**	**-106.5**	**-108.7**
today one year ago i lost a neighbour and friend too so here ls a quote that got me through my worst times	-135.13	-126.2	-125.11
today one year ago i lost a neighbovr and friend too so here is a quote that got me through my worst times	-136.53	-131.82	-130.4
today one year ago i lost a neighbovr and friend too so here ls a quote that got me through my worst times	-156.89	-151.49	-146.67
today one year ago i lost a nelghbour and friend too so here is a quote that got me through my worst times	-143.87	-137.27	-134.91
today one year ago i lost a nelghbour and friend too so here ls a quote that got me through my worst times	-164.07	-155.2	-151.04
today one year ago i lost a nelghbovr and friend too so here is a quote that got me through my worst times	-151.97	-147.98	-149.06
today one year ago i lost a nelghbovr and friend too so here ls a quote that got me through my worst times	-171.75	-166.26	-165.31
today one year ago l lost a neighbour and friend too so here is a quote that got me through my worst times	-125.83	-120.52	-119.66
today one year ago l lost a neighbour and friend too so here ls a quote that got me through my worst times	-145.9	-138.75	-135.2
today one year ago l lost a neighbovr and friend too so here is a quote that got me through my worst times	-147.04	-146.24	-140.2
today one year ago l lost a neighbovr and friend too so here ls a quote that got me through my worst times	-167.19	-164.26	-155.58
today one year ago l lost a nelghbour and friend too so here is a quote that got me through my worst times	-154.02	-150.67	-144.47
today one year ago l lost a nelghbour and friend too so here ls a quote that got me through my worst times	-174.51	-167.43	-159.2
today one year ago l lost a nelghbovr and friend too so here is a quote that got me through my worst times	-162.96	-161.59	-158.82
today one year ago l lost a nelghbovr and friend too so here ls a quote that got me through my worst times	-182.82	-178.72	-173.65

Tabla 5.4: Resumen de conversión de *leet* a texto en claro para tuits con tendencias suicidas

Tuit *leet*	Total de oraciones	Distil GPT2	GPT2	GPT2 Medium	Tuit Obtenido	¿Es igual al original?
0n3 d@y 111 k111 mys3lf @nd my sv1c1d3 n0t3 w111 jvst b3 s0m3 sh1t 11k3 my @rms @r3 k1nd@ chvbby By3	3	-133.02	-122.41	-121.28	one day ill kill myself and my suicide note will just be some shit like my arms are kinda chubby bye	Sí
Th1nk1ng 1n Sv1c1d3 h@s n0w b3c0m3 my n3w w0rld @nd 1 @m d3sp3r@t3ly s3@rch1ng f0r @nsw3rs	2	-85.65	-82.84	-90.32	thinking in suicide has now become my new world and i am desperately searching for answers	Sí
T0d@y 0n3 y3@r @g0 1 l0st @ n31ghb0vr @nd fr13nd t00 S0 h3r3 1s @ qv0t3 th@t g0t m3 thr0vgh my w0rst t1m3s	16	-114.62	-106.52	-108.76	today one year ago i lost a neighbour and friend too so here is a quote that got me through my worst times	Sí
1 n33d 4 gv1d4nc3 m3nt0r f4th3r f1gvr3 gv4rd14n 4ng3l 0r wh4t3v3r fvck 0vt th3r3 th4t c4n t3ll m3 wh4t t0 d0 w1th m3	2	-133.42	-124.55	-118.27	i need a guidance mentor father figure guardian angel or whatever fuck out there that can tell me what to do with me	Sí
1m 4 fvck1ng lvn4t1c wh3n 1 g3t 4ngry 1ts s0 t3rr1bl3 1 n33d t0 w0rk 0n th1s s0 b4d b3f0r3 1 3nd vp d01ng s0m3th1g 4g41nst m3	32	-141.98	-131.35	-129.33	im a fucking lunatic when i get angry its so terrible i need to work on this so bad before i end up doing somethig against me	Sí
Jvst w4nn4 fvck1ng sl33p 4nd n3v3r w4k3 vp 3v3r 4g41n	1	-	-	-	just wanna fucking sleep and never wake up ever again	Sí
1 fvck1ng m1ss y0v bvt y0vr3 pr0b4bly b3tt3r 0ff w1th0vt m3	4	-69.51	-57.37	-57.53	i fucking miss you but youre probably better off without me	Sí
m3 wh3n 1 f1n4lly d13 4ft3r s4y1ng 1 w4nt t0 d13 3v3ry s1ngl3 d4y	4	-66.38	-63.11	-62.35	me when i finally die after saying i want to die every single day	Sí

Finalizado este experimento, se aplicó la solución implementada, con los datos del peor resultado de los experimentos en la fase de comprobación. Estos son aquellos a los que se les realizó el reemplazo de todas las vocales por un carácter *leet* asociado a cada una de ellas.

En primer lugar, se ingresó el conjunto de datos al módulo implementado para obtener el total de frases posibles para cada uno de los 9135 tuits. El resumen del resultado del total de frases generadas, se presenta en la tabla 5.5.

Tabla 5.5: Total de frases generadas con el módulo implementado

Tuits	Frases por grupo	Frases Totales
40	1024	40960
57	512	29184
76	256	19456
125	128	16000
128	64	8192
206	32	6592
349	16	5584
421	8	3368
629	4	2516
866	2	1732
6238	1	6238

Debido a que un total de 6238 tuits obtuvieron solamente una frase, estas pasan directamente a ser la frase transformada final; mientras que 2897 tuits generaron desde 2 hasta un máximo de 1024 frases posibles para efectuar la sustitución del tuit original. De esta forma, se obtuvieron un total de 133584 frases que pasaron a ser analizadas por los tres modelos de lenguaje seleccionados, obteniendo así la calificación de cada una de ellas, seleccionando por último la frase adecuada para cada uno de los tuits que se esta procesando.

El proceso de efectuar el cálculo de la calificación para cada una de las oraciones por el modelo del lenguaje, fue la parte con la mayor duración y cambió según el modelo seleccionado. En esta ejecución, la duración de cada uno de los modelos fue el siguiente: el modelo DistilGPT2 tardó 41,81 horas; el modelo GPT2 tardó 42,4 horas y el modelo GPT2Medium tardó un total de 89.52 horas. Como resultado del proceso, se obtienen tres conjuntos de datos con los 9135 tuits mejor calificados por cada uno de los modelos.

La siguiente actividad, consistió en ingresar los tres conjuntos de datos al modelo

Tabla 5.6: Consolidado de resultados de experimentos

Conjunto de datos	*Accuracy*	*F1 score*	*Precision*	*Recall*	*ROC - AUC*	Tuits suicidas	Tuits no suicidas
Fase de entrenamiento	96.450%	96.240%	94.480%	98.060%	0.989	82	4145
Todas las vocales	51.990%	3.650%	25.540%	1.960%	0.806	4144	83
Distil-GPT2	96.220%	95.990%	94.460%	97.560%	0.988	103	4124
GPT2	96.320%	96.090%	94.470%	97.780%	0.989	94	4133
GPT2-Medium	96.440%	96.230%	94.480%	98.040%	0.989	83	4144

BERT obtenido en la fase de entrenamiento y se validó el resultado final del proceso de clasificación. Los resultados obtenidos se representan en la tabla 5.6 donde además del incluir los resultados de los tres conjuntos de datos obtenidos, se incluyeron las estadísticas del conjunto de datos de prueba original y el experimento con el conjunto de datos donde se modificaron todas las vocales, pasa así, efectuar la comparación de clasificación de tuits con tendencias suicidas también presentado en la figura 5.9.

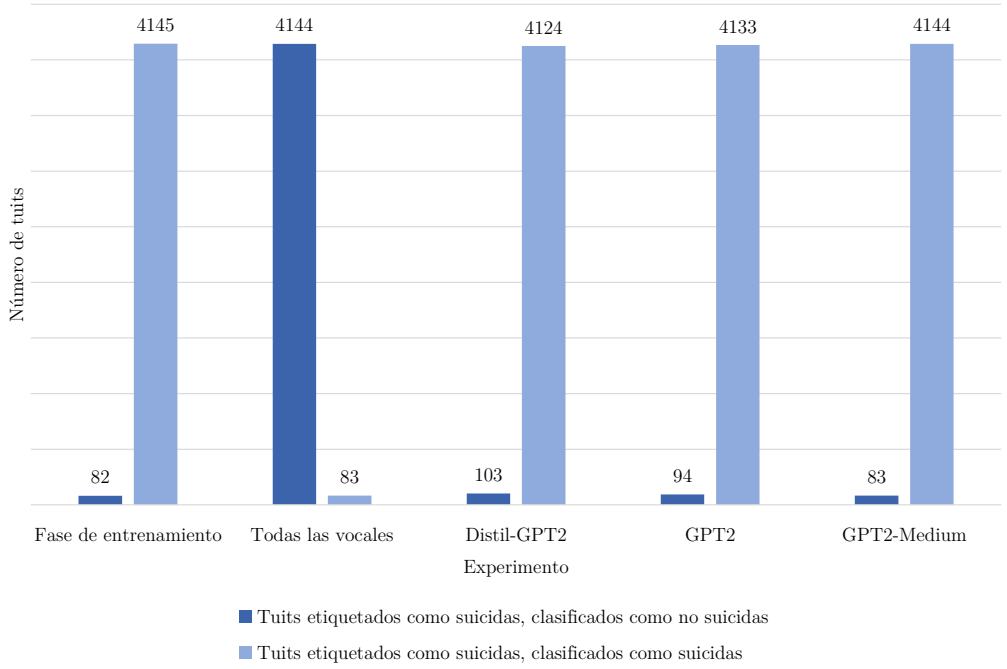


Figura 5.9: Resultados de clasificación consolidados

Capítulo 6

Discusión de los resultados

De acuerdo con los datos obtenidos en la fase cuatro, puede afirmarse que el resultado obtenido con los tres modelos para restaurar los tuits originales es bastante satisfactorio. Utilizando el modelo de lenguaje más sencillo, el DistilGPT2, genera una diferencia de 21 tuits con respecto a los datos de prueba originales. Con el modelo GPT2, la diferencia es de 12 tuits. Y con el modelo más robusto seleccionado, la diferencia es únicamente de un tuit mal clasificado.

De esta manera, para identificar por qué se está generando la diferencia en uno de los tuits, se realizó la búsqueda en el conjunto de datos original y el conjunto de datos del modelo GPT2Medium, con el objetivo de obtener los dos tuits que están generando la diferencia al momento de llevar a cabo la respectiva comparación, para poder concluir, si la conversión ejecutada afecta de manera negativa la clasificación, o si por el contrario, la etiqueta asignada al tuit original no se realizó adecuadamente y posterior a la modificación, el tuit se está asignando dentro de la categoría correcta como no suicida.

El tuit que genera la diferencia obtenido del conjunto de datos original es el siguiente:

suicide in the mountain west are high researcher are asking why rvd6lsfn3w

Y el tuit equivalente en el conjunto de datos GPT2Medium es:

suicide in the mountain west are high researcher are asking why rud6lsfnew

Comparando ambos tuits se identifica que la diferencia está en las ultimas palabras *rvd6lsfn3w* y *rud6lsfnew*. Esta palabra podría estar asociada a una incorrecta limpieza en el tuit ya que al parecer, puede estar asociado al autor que realizó la publicación del trino.

De acuerdo con las reglas implementadas en el módulo de conversión, la letra ´´v" y el número ´´3" fueron reemplazados por las letras ´´u" y ´´e" respectivamente. Analizando el contenido del texto, es posible concluir que el tuit no hace referencia a algún tipo de tendencia suicida por un usuario sino a algún evento que sucedió en algún lugar. Por lo que el cambio en esta única palabra, alteró la etiqueta de clasificación asociada al tuit original de suicida a no suicida. Así se confirma nuevamente la hipótesis que la combinación de caracteres numéricos y alfanuméricos, generan una alta variabilidad al momento de ejecutar una clasificación, inclusive, con palabras que no tienen sentido alguno como la identificada. Por ello, aunque la etiqueta de este tuit estaba asociada a uno de tendencia suicida, es valido concluir que el cambio en la palabra por el módulo implementado mejoró la clasificación del tuit e inclusive lo calificó como no suicida, que podría ser la verdadera etiqueta que se puede asignar a este tuit en particular.

Parte V

Conclusiones y trabajo futuro

Capítulo 7

Conclusiones

- El modelo de transformadores y los mecanismos de atención que utilizan, demuestran ser una de las herramientas recientes que presentan los mejores resultados en el procesamiento de tareas asociadas al tratamiento del lenguaje natural, disminuyendo el número de transformaciones requeridas sobre las secuencias de entrada, logrando extraer todas las características y modelarlas de una manera precisa para las diferentes tareas de clasificación. Para este estudio en particular, el modelo BERT genero excelentes métricas de desempeño en la detección de tendencias suicidas en tuits compartidos por los usuarios de la red social *twitter* generando un modelo acertado para utilizar en las demás tareas de esta investigación.

- Las practicas de modificación de contenido mediante diferentes técnicas de *leet speaking*, afectaron de manera negativa el desempeño del modelo tipo transformador BERT entrenado para la tarea de clasificación y detección. Con las tareas realizadas, fue posible demostrar que la modificación en tuits alusivos a compartir tendencias suicidas con esta técnica, ocasionaron que las métricas del clasificador fueran bastante negativas las cuales podrían conllevar al replanteamiento o rechazo del modelo entrenado.

- Al ser el *leet speaking* una tendencia relativamente nueva pero cuyo uso se ha expandido de una manera bastante rápida principalmente en usuarios jóvenes en su interacción diaria en las redes sociales, se ha convertido en una variable que es actualmente ignorada en muchos de los temas de investigación en las tareas de procesamiento de lenguaje natural, pero que como pudo ser evidenciado, genera impacto negativo en las tareas de procesamiento de textos escritos por los usuarios. Es por

ello que es importante realizar la inclusión de esta variable en el diseño e implementación de las nuevas investigaciones, o recurrir a herramientas adicionales que permitan manejar este tipo de tendencia para que su ocurrencia no interfieran en las demás fases de los diferentes proyectos que analicen estudios de comportamiento de usuarios en las redes sociales.

- El prototipo implementado haciendo uso de modelos de lenguaje que cuentan con un mayor número de parámetros y un mayor conjunto de datos utilizados en su entrenamiento, permitió que la transformación realizada a cada una de las publicaciones realizadas y que la puntuación obtenida para cada una de las opciones de sustitución, se realizara de una manera más precisa conforme a las reglas de escritura del idioma inglés. De esta manera, se confirmó que entre más robusto fue el modelo seleccionado, más se aproximó el tuit seleccionado al tuit original, realizando así una clasificación precisa llegando inclusive a mejorar el resultado de clasificación que el modelo inicial entrenado y realizando la correcta detección de la tendencia del tuit.

- Al comparar los resultados de la solución implementada, se identificó que el modelo realizó la transformación de los tuits en formato *leet speaking* manteniendo intacta su intención y sin requerir que los modelos de clasificación originales requirieran de algún proceso de modificación para colocar la etiqueta correcta para la tendencia detectada. Por lo tanto, se concluye que el prototipo propuesto en esta tesis es una buena alternativa para solucionar el problema de investigación.

Capítulo 8

Trabajo futuro

Al finalizar el proyecto, se identifican las siguientes tareas como posible trabajo a futuro:

- Técnicas de *leet speaking* fuera del alcance del estudio actual. Aunque las técnicas sencillas aún son las más predominantes en las diferentes publicaciones realizadas por los usuarios, día a día se observan nuevas formas en que los usuarios realizan modificaciones cuando quieren publicar algún tipo de contenido que va en contra de las políticas de comportamiento de las diferentes redes sociales. Dentro de las nuevas técnicas de modificación se encuentran aquellas en donde la publicación de mensajes textuales es realizada en archivos de imagen, las cuales son modificadas para ocultar las palabras asociadas a la violación de las reglas de comportamiento y después son compartidas como un archivo adjunto. En la figura8.1 se presenta un ejemplo de la inclusión de texto en imágenes y su alteración.

  Otra de las técnicas utilizadas, es la inclusión de símbolos adicionales en las palabras, separándola en partes que de cara a su análisis por parte de los filtros de moderación no genera ninguna alerta, pero de cara a los usuarios es sencillo de realizar la interpretación de la intención que se desea transmitir con dicha publicación. Un ejemplo de esta técnica se presenta en la figura 8.2

- Modelos para la detección de otro tipo de emociones. Aunque el modelo fue probado únicamente para la detección de tendencias suicidas, el componente utilizado también podría aplicarse en otro tipo de modelos donde se utilice el *leet speaking* como método de alteración del texto original, como por ejemplo en acoso, racismo, insultos o cualquier otro tipo de sentimientos que se encuentren en contra de las normas de la

Figura 8.1: Técnica de modificación de texto en imágenes.

Figura 8.2: Técnica de modificación incluyendo caracteres adicionales en palabras.

comunidad de cada una de las redes sociales.

- Publicaciones realizadas en otras redes sociales. Debido a que cada una de las redes maneja sus propias normas para los usuarios, es usual que en otras redes sociales se utilicen también las técnicas de *leet speaking* en contenidos textuales que suelen ser más largos que un trino. Es por ello, que analizar el componente implementado con nuevas reglas de conversión y con orígenes textuales más largos, es una nueva forma para verificar su comportamiento y desempeño.

- Modelos de lenguaje entrenados con datos de la red a analizar. Es conveniente realizar la búsqueda, encontrar e integrar modelos de lenguaje cuyo entrenamiento

haya sido realizado con datos específicos sobre cada una de las redes sociales a las que se esté realizando el análisis de su contenido. De esta forma, podría realizarse una calificación más acertada de las opciones a sustituir una publicación original conforme a la forma en que se escriben las publicaciones en cada una de las redes sociales, debido a que la limitante de número de caracteres en el caso de *twitter*, cambia la forma en que son escritas las publicaciones en comparación a *Facebook* donde una publicación cuenta con un límite mucho mayor con respecto al número de caracteres que pueden ser incluidos.

Bibliografía

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," pp. 5998–6008, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[3] L. Y. Leong, T. S. Hew, K. B. Ooi, V. H. Lee, and J. J. Hew, "A hybrid SEM-neural network analysis of social media addiction," *Expert Systems with Applications*, vol. 133, pp. 296–316, 2019.

[4] A. Dhir, Y. Yossatorn, P. Kaur, and S. Chen, "Online social media fatigue and psychological wellbeing—A study of compulsive use, fear of missing out, fatigue, anxiety and depression," *International Journal of Information Management*, vol. 40, no. January, pp. 141–152, 2018.

[5] M. Tateno, A. R. Teo, W. Ukai, J. Kanazawa, R. Katsuki, H. Kubo, and T. A. Kato, "Internet addiction, smartphone addiction, and hikikomori trait in Japanese young adult: Social isolation and social network," *Frontiers in Psychiatry*, vol. 10, no. JULY, pp. 1–11, 2019.

[6] B. Rimé, "Emotion Elicits the Social Sharing of Emotion: Theory and Empirical Review," *Emotion Review*, vol. 1, no. 1, pp. 60–85, 2009.

[7] E. D. Klonsky, A. M. May, and B. Y. Saffer, "Suicide, suicide attempts, and suicidal ideation," *Annual Review of Clinical Psychology*, vol. 12, no. 1, pp. 307–330, 2016.

[8] S. M. Dunlop, E. More, and D. Romer, "Where do youth learn about suicides on the Internet, and what influence does this have on suicidal ideation?," *Journal of Child Psychology and Psychiatry and Allied Disciplines*, vol. 52, no. 10, pp. 1073–1080, 2011.

[9] J. L. Jasso-Medrano and F. López-Rosales, "Measuring the relationship between social media use and addictive behavior and depression and suicide ideation among university students," *Comput. Hum. Behav.*, vol. 87, pp. 183–191, 2018.

[10] J. L. Oliffe, J. S. Ogrodniczuk, J. L. Bottorff, J. L. Johnson, and K. Hoyak, ""You feel like you can't live anymore": Suicide from the perspectives of Canadian men who experience depression," *Social Science and Medicine*, vol. 74, no. 4, pp. 506–514, 2012.

[11] E. Aboujaoude, "Rising suicide rates: an under-recognized role for the internet?. world psychiatry : official journal of the world psychiatric association (wpa)," *Cjournal of the World Psychiatric Association (WPA)*, vol. 15(3), pp. 225–227, 2016.

[12] H. A. e. a. DeSmet A., Deforche B., "Traditional and cyberbullying victimization as correlates of psychosocial distress and barriers to a healthy lifestyle among severely obese adolescents a matched case control study on prevalence and results from a cross sectional study," *Cjournal of the World Psychiatric Association (WPA)*, vol. 14, p. 224, 2016.

[13] U. Tambe, N. Kakad, S. Suryawanshi, and S. Bhamre, *Content Filtering of Social Media Sites Using Machine Learning Techniques*. 12 2021.

[14] M. Kowalska and M. Wróbel, "Basic Emotions," 2017.

[15] H. Kellerman, "EMOTION: Theory, Research, and Experience," *The Measurement of Emotions*, vol. 1, p. ii, 1989.

[16] B. N. Colby, A. Ortony, G. L. Clore, and A. Collins, *The Cognitive Structure of Emotions.*, vol. 18. 1989.

[17] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[18] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text: Machine learning for text-based emotion prediction," *HLT/EMNLP 2005 - Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, no. October, pp. 579–586, 2005.

[19] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4629 LNAI, no. June, pp. 196–205, 2007.

[20] K. R. Scherer and H. G. Wallbott, "Evidence for universality and Cultrual Variation," vol. 66, no. 2, pp. 310–328, 1994.

[21] C. Strapparava and R. Mihalcea, "SemEval-2007 task 14: Affective text," *ACL 2007 - SemEval 2007 - Proceedings of the 4th International Workshop on Semantic Evaluations*, no. June, pp. 70–74, 2007.

[22] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 Task 1: Affect in Tweets," pp. 1–17, 2018.

[23] A. Chatterjee, K. N. Narahari, M. Joshi, and P. Agrawal, "SemEval-2019 Task 3 : EmoContext Conte[1] A. Chatterjee, K. N. Narahari, M. Joshi, and P. Agrawal, "SemEval-2019 Task 3 : EmoContext Contextual Emotion Detection in Text," pp. 39–48, 2019.xtual Emotion Detection in Text," pp. 39–48, 2019.

[24] P. Patwa, G. Aguilar, S. Kar, S. Pandey, S. PYKL, B. Gambäck, T. Chakraborty, T. Solorio, and A. Das, "SemEval-2020 Task 9: Overview of Sentiment Analysis of Code-Mixed Tweets," 2020.

[25] G. Sidorov, S. Miranda-Jiménez, F. Viveros-Jiménez, A. Gelbukh, N. Castro-Sánchez, F. Velásquez, I. Díaz-Rangel, S. Suárez-Guerra, A. Treviño, and J. Gordon, "Empirical study of machine learning based approach for opinion mining in tweets," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7629 LNAI, no. PART 1, pp. 1–14, 2013.

[26] I. D. Rangel, G. Sidorov, and S. S. Guerra, "Creación y evaluación de un diccionario marcado con emociones y ponderado para el español," *Onomazein*, vol. 29, no. 1, pp. 31–46, 2014.

[27] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning: Machine Learning Book," p. 785, 2016.

[28] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," 2017.

[29] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," 2017.

[30] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," 2017.

[31] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," 2022.

[32] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, Nov. 2016.

[33] I. K. S. Al-Tameemi, M.-R. Feizi-Derakhshi, S. Pashazadeh, and M. Asadpour, "A comprehensive review of visual-textual sentiment analysis from social media networks," 2022.

[34] E. Villatoro-Tello, G. Ramírez-De-la rosa, and H. Jiménez-Salazar, "UAM's participation at CLEF eRisk 2017 task: Towards modelling depressed bloggers," *CEUR Workshop Proceedings*, vol. 1866, 2017.

[35] S. H. Hosseini-Saravani, S. Besharati, H. Calvo, and A. Gelbukh, "Depression Detection in Social Media Using a Psychoanalytical Technique for Feature Extraction and a Cognitive Based Classifier," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12469 LNAI, pp. 282–292, 2020.

[36] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Detection of suicide ideation in social media forums using deep learning," *Algorithms*, vol. 13, no. 1, pp. 1–19, 2020.

[37] R. Sawhney, H. Joshi, S. Gandhi, and R. R. Shah, "A Time-Aware Transformer Based Model for Suicide Ideation Detection on Social Media," pp. 7685–7697, 2020.

[38] T. Zhang, A. M. Schoene, and S. Ananiadou, "Automatic identification of suicide notes with a transformer-based deep learning model," *Internet Interventions*, vol. 25, 9 2021.

[39] Q. un Nisa, R. Muhammad, and S. L. T. Karachi, "Towards transfer learning using bert for early detection of self-harm of social media users."

[40] F. Haque, R. U. Nur, S. A. Jahan, Z. Mahmud, and F. M. Shah, "A transformer based approach to detect suicidal ideation using pre-trained language models," Institute of Electrical and Electronics Engineers Inc., 12 2020.

[41] R. Sawhney, H. Joshi, S. Gandhi, and R. R. Shah, "A time-aware transformer based model for suicide ideation detection on social media."

[42] R. Sawhney, H. Joshi, A. Nobles, and R. R. Shah, "Towards emotion- and time-aware classification of tweets to assist human moderation for suicide prevention," 2021.

[43] P. Santos and A. Sánchez, "El inglés y el español desde una perspectiva cuantitativa y distributiva: equivalencias y contrastes," *Estudios Ingleses de la Universidad Complutense*, vol. 19, no. 0, pp. 15–44, 2011.