



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Chatbot wattson entrenado para la detección de
emociones**

TESIS

Que para obtener el título de
Ingeniero en Computación

PRESENTA

Jorge Luis Ramos Barraza

DIRECTOR DE TESIS

Dr. Iván Vladimir Meza Ruiz



Ciudad Universitaria, Cd. Mx., 2022

Resumen

Todos los días experimentamos diversas emociones, desde aquellas que causan un sentimiento de felicidad hasta aquellas que nos hacen experimentar miedo; El psicólogo Paul Ekman, llegó a la conclusión de que existen seis emociones universales, (i) *alegría*, (ii) *ira*, (iii) *miedo*, (iv) *asco*, (v) *sorpresa* y (vi) *tristeza* las cuales serán identificadas a partir de un texto que se ingrese al modelo que se entrenó en este trabajo. Se presenta el proceso que se llevó a cabo para poder entrenar un modelo etiquetador de flujo de las emociones ya mencionadas, desde la obtención de un corpus en idioma español y la etiquetación de forma manual del mismo con ayuda de la herramienta Doccano, hasta el proceso de entrenamiento con dos diferentes modelos de lenguaje BERT y electra, también se presenta la explicación de la arquitectura utilizada la cual es el *transformer* y se realizaron diferentes experimentos en los cuales se variaron los parámetros de entrenamiento hasta obtener los mejores resultados, con el modelo de lenguaje BERT y 100 épocas de entrenamiento llegando se obtuvo un modelo con una precisión del 99.3%, cuando se es configurado en modo evaluación se obtiene una precisión del 72.8%, también se evaluó el porcentaje de precisión con cada una de las 6 emociones diferentes llegando a ser el asco la emoción que mejor logra etiquetar el modelo y la sorpresa aquella peor identifica.

Capítulo 1	3
Introducción	3
1.1 Objetivo	3
1.2 Alcance	3
1.3 Metas	3
1.4 Justificación	3
1.5 Antecedentes	4
1.6 Metodología	4
1.7 Programa de actividades	5
Capítulo 2	6
Estado del arte	6
2.1 Chatbots	6
2.2 Emociones	7
2.3 Chatbots y emociones	7
2.4 Modelos de lenguaje	9
Capítulo 3	10
Arquitectura Transformer	10
3.1 Antecedentes de la arquitectura	10
3.2 Funcionamiento	11
3.3 Codificador	11
3.4 Decodificador	17
3.5 BERT	20
Capítulo 4	24
Datos	24
4.1 Origen de los datos	24
4.2 Anonimización	24
4.2 Etiquetado	27
4.3 Stop Words	31
4.4 Análisis de los datos	32
Capítulo 5	38
Pruebas y Resultados	38
Capítulo 6	47
Conclusiones	47
Fuentes de información	49

Capítulo 1

Introducción

En este capítulo se presentarán los aspectos más importantes sobre la tesis, como lo son; el objetivo, el alcance, las metas, la justificación, los antecedentes, la metodología, el programa de actividades, el calendario de actividades así como la principal bibliografía a utilizar, con la finalidad de que el lector entienda cuál es el camino que va a seguir a lo largo de este trabajo y a donde va a llegar.

1.1 Objetivo

Entrenar un modelo *transformer* capaz de identificar el flujo emocional que existe en las vivencias de una persona que decide compartirlas a través de un chatbot llamado Wattson, evaluando su desempeño para obtener un alto grado de precisión.

1.2 Alcance

Este trabajo se limita a mostrar el proceso que se llevó a cabo para entrenar un modelo etiquetador de flujo emocional.

1.3 Metas

Las metas a lograr son la obtención de datos que sirvan para entrenar el modelo como lo son historias y entradas de diario; la anonimización de estas historias, el etiquetado de los datos y el entrenamiento del modelo.

1.4 Justificación

Las personas suelen compartir con otras personas ciertas vivencias, dependiendo el nivel de confianza que exista entre estas, será el nivel emotivo que la historia contenga. Si una persona compartiera estas mismas vivencias con un chatbot, podría plasmar completamente sus emociones en cada una de ellas, al desarrollar este trabajo se espera que la interacción de la persona sea lo más natural posible, es decir no ser tan estricto con las reglas ortográficas, por ejemplo que el modelo sea capaz de identificar textos como el siguiente *“el día de hoy me siento bien porque saque 10 en mi examen de historia, fui el que tuvo la calificación mas alta y mis compañeros me felicitaron”* y pueda ser etiquetado de forma correcta. En la actualidad *transformer* es la arquitectura más efectiva para el procesamiento de lenguaje natural, la arquitectura trabaja básicamente con una entrada, que es procesada en un codificador y un decodificador que dan como resultado la salida del

texto etiquetado para fines de este trabajo. En este trabajo se utilizará un modelo en idioma español para detectar el flujo emocional que existe en las entradas del chatbot.

1.5 Antecedentes

Como base y antecedente de este trabajo se utilizará el Procesamiento de Lenguaje Natural (PLN) campo de la inteligencia artificial y de la lingüística que persigue el estudio de la interacción que hay entre el lenguaje humano y las computadoras. Un paper de google intitulado "Attention is All You Need" ¹ presentó la arquitectura del *transformer*, marcando un hito importante en el procesamiento de lenguaje natural, pues presenta una arquitectura nueva y eficaz. En particular, se utilizará el modelo BERT debido a la capacidad que tiene de entender el contexto de una entrada y de esta forma ofrecer resultados más acordes para el objetivo de este trabajo. Un artículo intitulado "Sentiment analysis of informal textual communication in cyberspace"² trabajó de igual forma con analizar las emociones que hay en las entradas de texto que los usuarios de videojuegos realizan, de esta forma buscaron representar la correcta interpretación de esa emoción mediante lenguaje corporal que el personaje del videojuego mostraba al jugador.

1.6 Metodología

El lenguaje natural así como la tecnología es algo con lo que inevitablemente todos los días interactuamos, hoy en día es posible comunicarse con un chatbot de manera que al usuario le sea casi imperceptible saber que se trata de un bot y no de una persona. Crear un chatbot y entrenarlo mediante inteligencia artificial para que sea capaz de identificar emociones es un proceso que requiere del conocimiento de lenguaje natural así como de programación, es por esto que como método de este trabajo se utilizará el procesamiento de lenguaje natural así como la arquitectura *transformer*, ya que son los mejores caminos que se pueden seguir para dar con el objetivo del trabajo, actualmente sigue siendo un reto aplicar estas tecnologías para poder construir un modelo robusto y eficiente debido a que se requieren conocimientos más profundos del lenguaje así como de inteligencia artificial, debido a esto se pondrán a prueba todas las habilidades obtenidas durante la formación de un ingeniero en computación realizando de esta forma un aporte a la investigación y desarrollo de chatbots en el idioma español, dejando de esta forma un antecedente para trabajos futuros que requieran de esta información con fines educativos o de investigación.

¹ Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. ArXiv:1706.03762 [Cs]. <http://arxiv.org/abs/1706.03762>

² Paltoglou, G., Gobron, S., Skowron, M., Thelwall, M., & Thalmann, D. (s/f). Sentiment analysis of informal textual communication in cyberspace. 16.

1.7 Programa de actividades

La obtención de datos será por medio de una encuesta realizada mediante google, misma que será pública y abierta, a fin de obtener un corpus necesario que sirva para entrenar el modelo, la anonimización de estos datos consiste en quitar la información sensible de los mismos para después ser etiquetados, se experimentará con los datos para alimentar el sistema de etiquetado y por último se entrenará un modelo usando arquitectura *transformer*.

Calendario de Actividades

Actividad/Mes	1	2	3	4	5	6
Obtencion de Datos						
Anonimizacion de Datos						
Etiquetacion de datos						
Entrenamiento del modelo						
Construccion de la tesis						

Bibliografía

Artículo con la metodología principal: Transformer

* Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. NIPS.

Artículo donde la metodología se aplica a sistemas de dialogos/chatbots

* Chen, Y.-N., Celikyilmaz, A., & Hakkani-Tur, D. (2018). Deep learning for dialogue systems. Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts, 25–31.

Libros de texto del campo

* Jurafsky, D., & Martin, J. H. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2. ed., Pearson internat. ed). Prentice Hall, Pearson Education Internat.

* Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT Press

Capítulo 2

Estado del arte

Este capítulo es una introducción al presente trabajo, aquí el lector aprenderá conceptos y ejemplos sobre chatbots, emociones y procesamiento del lenguaje natural que le ayudarán a comprender de una mejor manera capítulos posteriores.

2.1 Chatbots

Un chatbot es un programa que tiene como principal objetivo proveer de respuestas eficaces sobre temas específicos a sus usuarios. Su historia tiene origen en el año 1964 con la creación de Eliza, un chatbot diseñado para que actuara como un psicólogo el cual fue desarrollado por Joseph Weizenbaum (Weizenbaum, 1976, "ELIZA", 2021); unos años más tarde en 1972 surge Parry, un chatbot que simulaba ser una persona con esquizofrenia, desarrollado por el psiquiatra Kenneth Colby ("PARRY", 2021); Alice, un chatbot muy importante y distinguido debido a que contaba con la implementación de procesamiento de lenguaje natural desarrollado originalmente por Richard Wallace en 1995 (AbuShawar & Atwell, 2015) ("Artificial Linguistic Internet Computer Entity", 2020); Simi Simi nacido en el año 2002, es un chatbot conversacional desarrollado con el uso de inteligencia artificial que le permite aprender más y más cada vez que se interactúa con él (Olivas, 2016); watson por parte de IBM apareció en el año 2011 con la implementación de inteligencia artificial que le permite crear, entrenar e implementar respuestas para responder a cualquier conversación sin importar la aplicación (IBM, 2021); en 2016 Facebook hace posible el desarrollo de chatbots en messenger.

Es muy común utilizar el término chatbot y asistente virtual como iguales sin embargo la diferencia radica en las capacidades de cada uno, mientras un chatbot está limitado a dar respuestas, los asistentes virtuales realizan tareas, los chatbots se enfocan en un área de aplicación, los asistentes no están limitados por ello, algunos asistentes personales como Siri, Alexa y el asistente de Google van más allá de brindar respuestas a sus usuarios, pero no por ello los chatbots dejan de ser interesantes, tal es el caso de "Victoria" un chatbot desarrollado por el gobierno de la Ciudad de México para mantener informados a los ciudadanos respecto a la pandemia del virus Covid-19 cuyo enfoque está dirigido a trámites y servicios para los ciudadanos de la capital (Agencia Digital de Innovación Pública, 2020). INTEBOT es un chatbot enfocado en relaciones comerciales que ayuda a sus usuarios a obtener un comprobante de pago o consultar calificaciones mediante el chatbot INTEBOT desarrollado por la empresa InteVolution incubada en la UNAM (UNAM, 2020).

Los sistemas de diálogo manejan 3 componentes básicos que son el reconocimiento automático de voz (ASR por sus siglas en inglés), la comprensión del lenguaje (LU por sus siglas en inglés), la gestión de diálogo (DM por sus siglas en inglés) arquitectura que se muestra en la figura 1³. Este trabajo está focalizado en la comprensión del lenguaje.

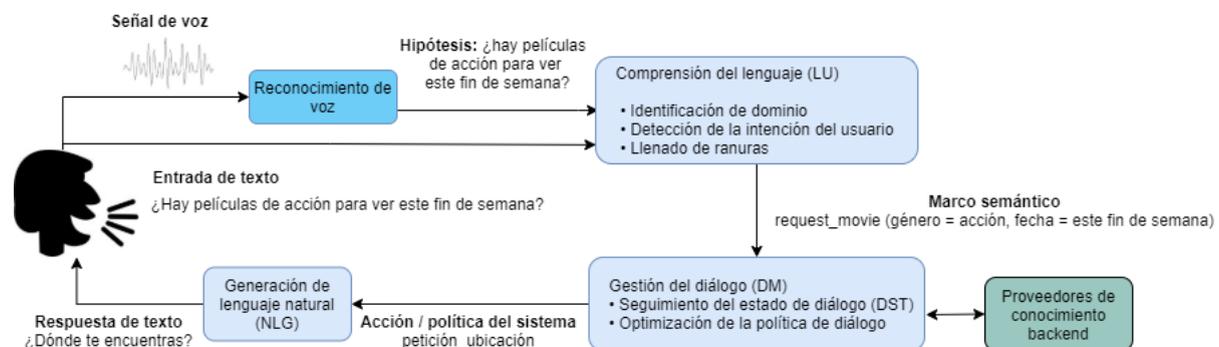


Figura 1: Diagrama de un sistema de diálogo (Tur & De Mori, 2011)

2.2 Emociones

En las ciencias sociales el estudio sobre las emociones es un área de estudio compleja, por ello utilizaremos como base el trabajo de Paul Ekman, psicólogo que se dedicó al estudio de las emociones, a lo largo de su vida realizó varios estudios donde a partir de las expresiones faciales llegó a la conclusión de que existen seis emociones universales para el ser humano que son; (i) *alegría*, (ii) *ira*, (iii) *miedo*, (iv) *asco*, (v) *sorpresa* así como (vi) *tristeza* (Ekman & Oster, 1981). Estas 6 emociones se consideran primarias o básicas, sin embargo existen emociones complejas las cuales son consideradas complementarias a las primarias (orientacionandujar, 2018) .

2.3 Chatbots y emociones

Al interactuar con un ordenador, en todo momento se intercambia información mediante ayuda del software, cuyo objetivo es hacer que este intercambio sea más eficiente, mejorando la experiencia de los usuarios a fin de minimizar errores y realizar tareas de forma más productiva (Hewett et al., 1992), para la comunicación con chatbots el intercambio de información puede ser texto o voz ya sea que el bot utilice tecnología de reconocimiento automático de voz (ASR por sus siglas en inglés) para recibir una entrada o texto a voz (TTS por sus siglas en inglés) como salida para el usuario o simplemente mediante texto (Asterisk, 2014). Woebot es un chatbot que trabaja con emociones, estableciendo un vínculo entre el usuario y el bot de tal manera que, al interactuar el usuario con el bot este se ve beneficiado de manera positiva por cuanto hace en aspectos de la

³ Traducida (Tur & De Mori, 2011)

salud mental, al mismo tiempo el bot aprende cada vez más y más sobre el usuario, funcionando de esta forma como un psicólogo autónomo el cual presenta excelentes resultados en la mayoría de los usuarios, se encuentra disponible para plataformas android e ios y funciona en idioma inglés (Woebot Health, 2021); Wysa es otro chatbot en idioma inglés que te ayuda a manejar tus emociones y pensamientos a través de distintas herramientas en las que se apoya, donde el usuario suele descargar toda esa carga emocional que lleva consigo y Wysa hace uso de inteligencia artificial para proporcionar respuestas acorde a la necesidad del usuario, escuchando a los usuarios como si se tratara de un terapeuta personal incluyendo un botón de ayuda en caso de alguna crisis emocional por lo que Wysa ha obtenido muy buenos resultados en sus usuarios (Wysa, 2021). Estos son solo 2 ejemplos de casos de éxito en los que los chatbots tienen un alto impacto en sus usuarios al manejar las emociones que se le transmiten, lo que hace notar la importancia del estudio. En la Figura 2 podemos observar un fragmento de la primera conversación que se tiene con cada uno de estos bots.

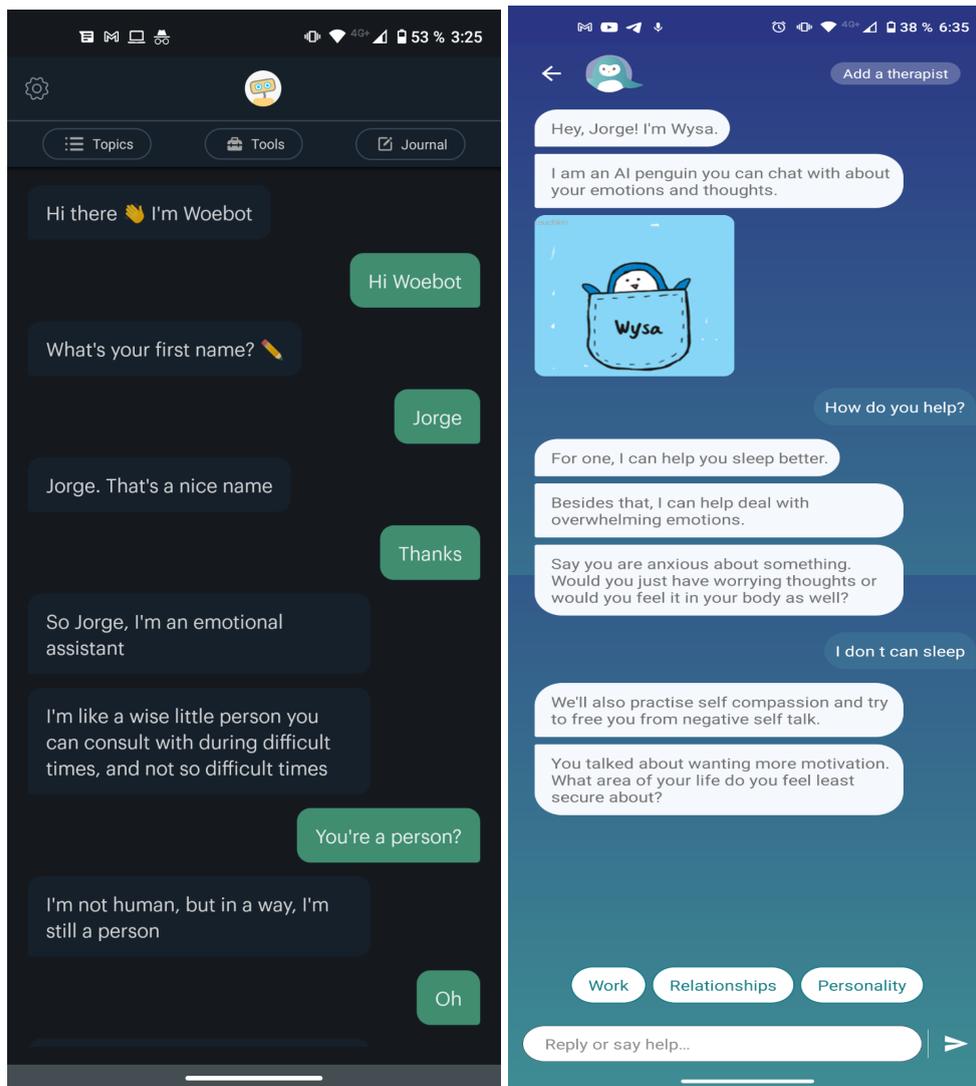


Figura 2: Primera conversación con Woebot y Wysa

2.4 Modelos de lenguaje

El Procesamiento de Lenguaje Natural (PLN) estudia la interacción entre el lenguaje humano y la computadora buscando que la comunicación entre el humano y la computadora sea lo más eficaz posible, para objeto de este trabajo es importante que el usuario comunique de manera correcta el mensaje que desea transmitir, para que sea interpretado de forma correcta por el modelo generado a través de la arquitectura *transformer*, el cual es el principal pilar para el desarrollo de este trabajo.

En este capítulo se mencionaron algunos de los chatbots más importantes que han existido a lo largo de la historia así como la relevancia de su estudio en armonía con las seis principales emociones que nuestro modelo aprenderá a identificar. Así mismo aunamos sobre la comunicación entre humanos y las computadoras y la importancia que representa en el día a día del ser humano, toda vez que los chatbots trabajan con emociones se vuelve un tema de interés, por último se revisó la importancia del procesamiento de lenguaje natural para este proyecto, en el siguiente capítulo se tocará más a fondo el tema de PLN con la arquitectura de *transformer*.

Capítulo 3

Arquitectura Transformer

En este capítulo se dará a conocer todo lo referente a la arquitectura *transformer*, se estudiarán sus antecedentes, su funcionamiento y por último el modelo de lenguaje BERT (Bidirectional Encoder Representations from Transformers).

3.1 Antecedentes de la arquitectura

Para entender la arquitectura transformer y su importancia, es necesario recordar cómo se hacían las cosas antes de este, sin ir muy atrás en un inicio se utilizaron redes neuronales multicapas para tareas básicas como la asociación de patrones, segmentación de imágenes o la compresión de datos por nombrar algunas tareas, posteriormente se utilizaron redes neuronales convolucionales derivadas de las redes neuronales multicapa con la diferencia de ser muy eficientes para tareas como procesar, analizar y comprender las imágenes para su clasificación y segmentación, más tarde surgieron las redes neuronales recurrentes (RNN), capaces de entender datos secuenciales, recordar las salidas anteriores de la secuencia como entrada, permitiendo que la red sea capaz de entender textos, sin embargo cuando el proceso de encadenamiento de salidas y entradas se hace muy largo, surge el problema de desvanecimiento de gradiente y como consecuencia las primeras entradas de la red dejan de tener relevancia para las entradas posteriores, convirtiendo el procesamiento de textos largos en un problema, sin embargo esto fue solucionado con la implementación de una arquitectura de memoria a corto plazo (en inglés LSTM) (Mehlig, 2021). Existieron más variantes de las RNN sin embargo fue en el año 2017 a partir de la necesidad de traducir de forma automática un texto de un idioma a otro cuando apareció la arquitectura del *transformer*, convirtiéndose en una alternativa para tareas de procesamiento de lenguaje natural, mejorando lo que las redes neuronales recurrentes ya hacían, esta nueva arquitectura adquirió gran popularidad debido a que implementa mecanismos de atención lo que le permite procesar texto de manera bidireccional además de una estructura de codificador y decodificador, gracias a esto se logró que las tareas sean paralelizables y no secuenciales como lo es con las redes neuronales recurrentes o convolucionales y por lo tanto sea más rápido de entrenar en comparación con otras arquitecturas, otra ventaja de esta arquitectura es que posee una memoria de largo plazo, lo que permite que no se pierda el contexto y la generación de textos largos sea más coherente. Poco tiempo después de la publicación del artículo surgieron modelos de lenguaje basados en el *transformer* como es el caso GPT (decodificador) GPT-2 y GPT-3 que utilizan la parte del codificador para realizar tareas como la clasificación o la generación

de lenguaje y un modelo muy particular llamado (Bidirectional Encoder Representations from Transformers) BERT o en español Representación de Codificador Bidireccional de Transformadores, el cual utiliza un codificador que servirá para el desarrollo de en este trabajo.

3.2 Funcionamiento

Para describir el funcionamiento de la arquitectura es mucho más sencillo partir de lo general a lo particular, pensemos entonces en el problema de traducción automática de textos de una forma muy general como se observa en la figura 3.



Figura 3: Diagrama general arquitectura transformer

En la figura 3 se observa del lado izquierdo una entrada de texto a la arquitectura en idioma francés, pasa por la arquitectura y como resultado obtenemos una salida con el mismo texto traducido al idioma inglés, esta es la representación sencilla pero visual de la arquitectura, pero, si empezamos a explorar lo que hay dentro del transformer veremos algo como se muestra en la figura 4

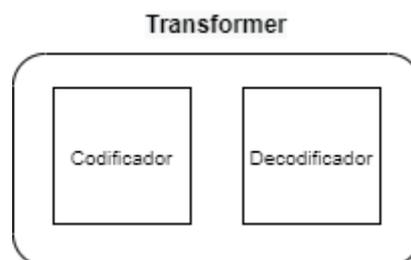


Figura 4: Arquitectura transformer encoder y decoder

En la figura 4 se muestran dos bloques que componen al *transformer*, llamados codificador y decodificador respectivamente, los cuales procesan la entrada y arrojan una salida con el resultado esperado.

3.3 Codificador

El codificador es una pila de "n" capas iguales donde cada una tiene dos subcapas. La primer subcapa es un mecanismo de auto-atención (self-attention) de múltiples cabezales (multi-head attention) figura 5⁴ color azul cian, la segunda es una red neuronal de

⁴ Traducida(Vaswani et al., 2017)

retroalimentación tal como se muestra en la figura 5 en color verde, como estas pueden existir 6 o “n” número de capas con la misma estructura que tiene la figura 5.

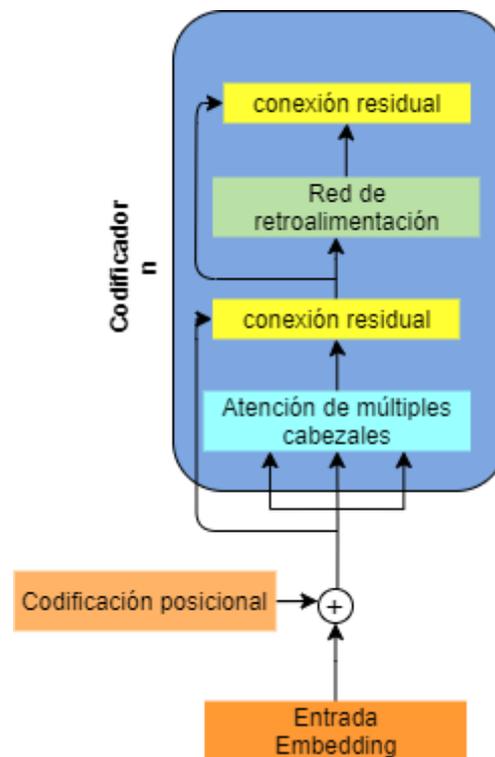


Figura 5: Arquitectura del codificador del transformer (Vaswani et al., 2017)

En la arquitectura del codificador se observan 5 bloques diferentes: fuera del codificador existe una entrada conocida como *embedding* y un codificador posicional de la misma entrada; dentro del codificador encontramos un bloque de atención de múltiples cabezales, conexiones residuales y una red de retroalimentación, a continuación se describe a mayor detalle cada uno de estos componentes.

- **Entrada Embedding:**

En este primer bloque se convierte cada palabra entrante a un vector de tamaño 512 también conocido como token, mismo que genera una representación numérica real para que la computadora lo pueda procesar, figura 6.

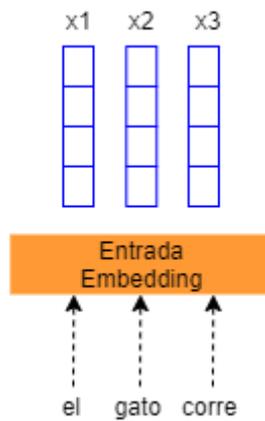


Figura 6: Entrada de texto

En la figura 6 el texto *el gato corre* es convertido a 3 vectores donde cada vector corresponde a una palabra de nuestro texto.

- **Codificación posicional:**

Este segundo bloque va a ser el encargado de conocer cuál es el orden correcto de cada uno de los vectores que se crearon en el bloque anterior, para ello utiliza una función senoidal para obtener vectores de codificación posicional (CP) de las posiciones pares, fórmula 1 y para las posiciones impares se utiliza la fórmula 2 que es una función cosenoidal donde “i” corresponde a la dimensión del modelo.

$$CP_{(posición,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (1)$$

$$CP_{(posición,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (2)$$

Una vez hemos obtenido la posición de nuestros vectores, realizamos una suma entre los tokens y los vectores de codificación posicional como se muestra en la figura 7 para obtener una matriz de vectores que contienen la posición de cada palabra.

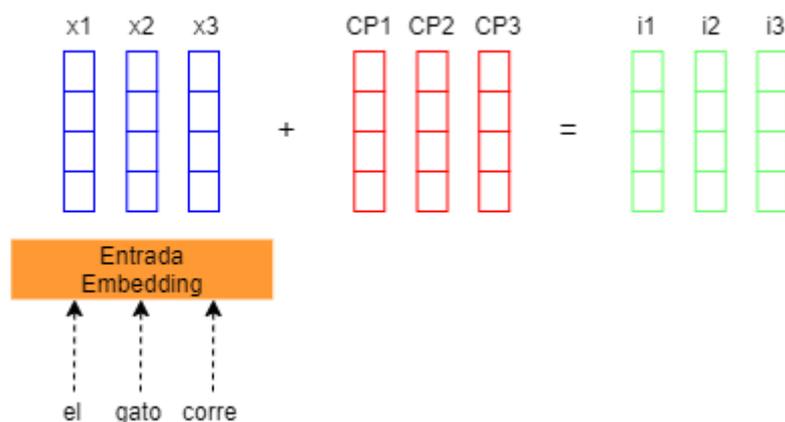


Figura 7: Resultado de la suma entre los vectores de tokens y de codificación posicional

Para entrar al siguiente bloque la matriz resultante realiza 4 copias de nuestra matriz resultante, de las cuales 3 irán al bloque de atención de múltiples cabezales y una a nuestro bloque residual.

- **Atención de múltiples cabezales:**

La atención de múltiples cabezales es la parte más importante del *transformer*, porque en esta parte se van a procesar nuestras matrices de vectores con el fin de establecer la relación que existe entre ellos, asignando un valor para saber a cuáles elementos se les tiene que prestar mayor atención para realizar nuestra tarea, en el ejemplo, la tarea es la traducción de un texto, para fines de este trabajo la tarea será la detección de emociones. Este bloque está compuesto por 3 pequeñas redes neuronales, cuyo objetivo es obtener 3 matrices (todas vienen de la misma) V, K y Q (Value, Key y Query) como se muestra en la figura 8⁵

⁵ Traducida (Vaswani et al., 2017)

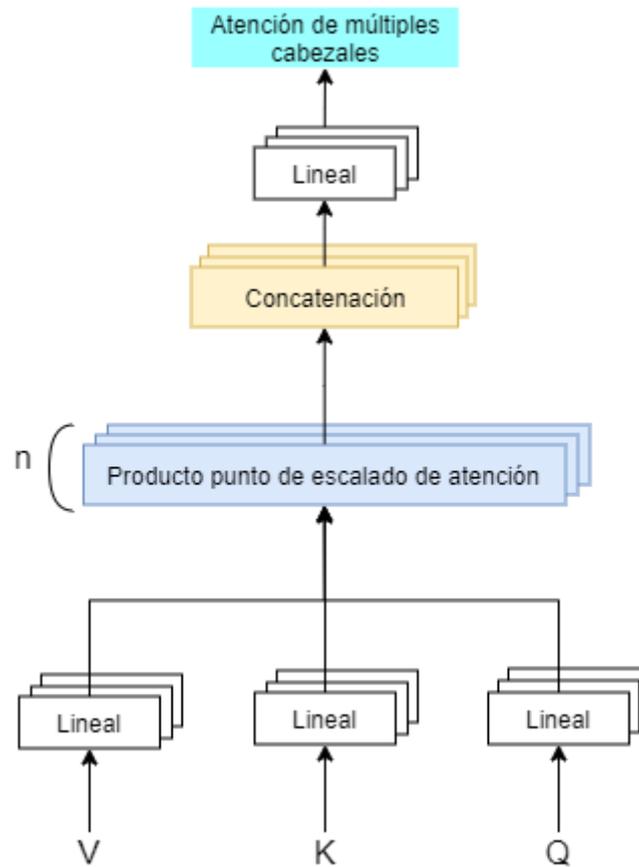


Figura 8: Diagrama interno del bloque de atención de múltiples cabezales (Vaswani et al., 2017)

Antes de entrar a detalle dentro de lo que ocurre en la atención de múltiples cabezales, es necesario entender qué es lo que ocurre dentro del producto punto de escalado de atención figura 9⁶

⁶ Traducida (Vaswani et al., 2017)

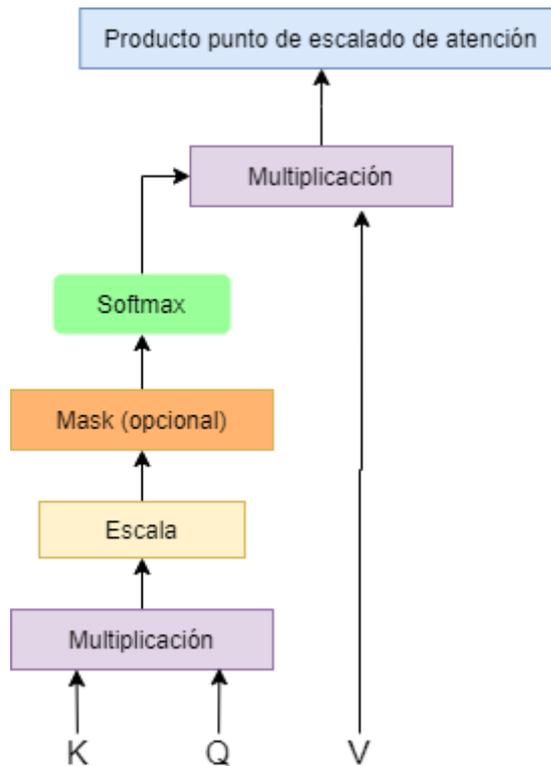


Figura 9: Producto punto de escalamiento de atención (Vaswani et al., 2017)

En la Figura 9 se muestra un diagrama con las matrices Value, Key y Query, que previamente pasaron por una capa lineal que transforma a las matrices para que se especialicen en cierta parte del lenguaje, por ejemplo en género, es por ello que para poder lograrlo se necesitan varias capas de atención como se muestra en la figura 8, estas matrices después serán modificadas mediante la fórmula 3, para ellos se aplica la transposición de K y se multiplica por Q de la forma QK^T de esta manera se obtiene el grado de asociación que existe entre las diferentes palabras, como resultado el modelo sabe a qué pares de palabras debe poner mayor atención y a cuáles no.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

Una vez realizada la multiplicación se debe escalar los resultados dividiéndolos entre las dimensiones del vector K de la siguiente forma $\frac{Qk^T}{\sqrt{d_k}}$, nos falta aplicarle una función softmax $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$ que será la encargada de representar nuestro resultado entre 0 y 1 donde si un par de palabras se encuentran más cercanas a 1 significa que existe una fuerte relación entre ellas y cuando se acerca a 0 significa que hay poca relación, por último se realiza una multiplicación entre la matriz

resultante después de aplicar la función softmax y la matriz V. Al final se obtiene una matriz que contiene nuevas representaciones de nuestras palabras como se observa en la figura 10.

	el	perro	esta	ladrando
el	0.2	0.1	0.1	0.23
perro	0.4	0.2	0.3	0.07
esta	0.03	0.1	0.2	0.24
ladrando	0.45	0.08	0.1	0.2

Figura 10: Matriz resultante del producto punto escalado de atención

Recordemos que fue necesario utilizar múltiples capas para poder obtener una especialización, por lo que es necesario concatenar nuestros resultados en una sola matriz para reducir la dimensionalidad y obtener una sola matriz manejable.

- **Conexión residual:**

El bloque residual cumple con una función muy importante, debido a las operaciones que se han realizado es posible que la información se degrade y por lo tanto el proceso de entrenamiento se vea afectado de forma negativa, es por ello que el bloque de conexión residual recibe como entrada dos matrices, a las cuales aplica primero una suma y posteriormente una normalización, de esta forma el peso de cada vector se procesa de forma más sencilla y al mismo tiempo se alimenta de la información original para recuperar el contexto.

- **Red de retroalimentación:**

Esta red se utiliza para transformar cada vector en una forma que sea digerible para el siguiente bloque.

Recordemos que contamos con un número n de codificadores con la misma estructura, es por ello que el proceso que acabamos de revisar se repite n número de veces para posteriormente entrar al decodificador.

3.4 Decodificador

El Decodificador de igual forma que el codificador es una pila de “n” número de capas, la arquitectura del decodificador se observa en la figura 11⁷, como se puede notar está

⁷ Traducido (Vaswani et al., 2017)

construido con los mismos bloques que ya revisamos con anterioridad exceptuando el bloque de atención de múltiples cabezales enmascarado, el cual veremos a continuación.

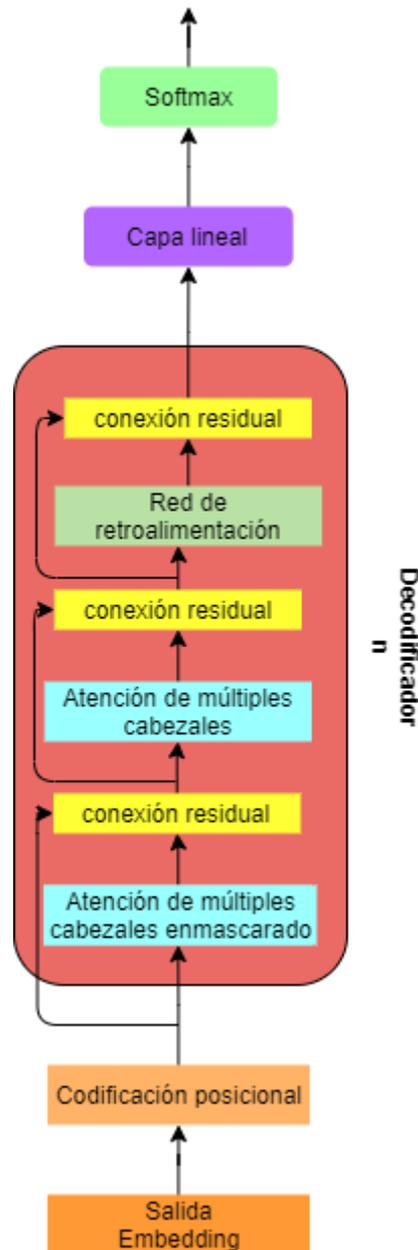


Figura 11: Arquitectura del decodificador del *transformer* (Vaswani et al., 2017)

Retomando el problema de traducción que se presentó al inicio de este capítulo, el decodificador en su bloque de salida *embedding* contiene la traducción de nuestro texto, pasa por un bloque de codificación posicional para conocer el orden de cada uno de los vectores, posteriormente entra al decodificador que se encuentra conectado al codificador,

de forma que le es posible conocer la información atencional de nuestro texto en el idioma original, permitiendo que el bloque de atención de múltiples cabezales contenga información de ambos textos, en las matrices *Query* y *Key* quedará la información del texto traducido, mientras que *Value* tendrá la información del texto original, de esta forma puede enfocar su atención en ambos textos. El proceso continúa y se repite con cada uno de los “n” decodificadores, por último la salida del codificador pasa por un bloque lineal, básicamente es una red neuronal que expande el vector en uno más grande con el número de palabras en idioma francés, finalizando el recorrido en bloque softmax que como ya vimos transforma cada palabra en una probabilidad, siendo la de mayor puntaje la salida, este proceso se repite con cada una de las palabras de nuestro texto de forma que el resultado es nuestro texto traducido. Podemos ver en la figura 12⁸ la arquitectura *transformer* completa.

Hemos revisado el caso en particular de traducción, pero esto no quiere decir que solo se utilice para traducir textos, otros enfoques que se le pueden dar a la arquitectura *transformer* es la generación de textos, clasificación de textos o imágenes, la sumarización, la respuesta a preguntas, reconocimiento de voz por mencionar algunas sin embargo se puede entrenar para realizar cualquier otra tarea relacionada con el procesamiento de lenguaje natural dependiendo de cual sea la tarea será la arquitectura que se utilizara.

⁸ Traducido (Vaswani et al., 2017)

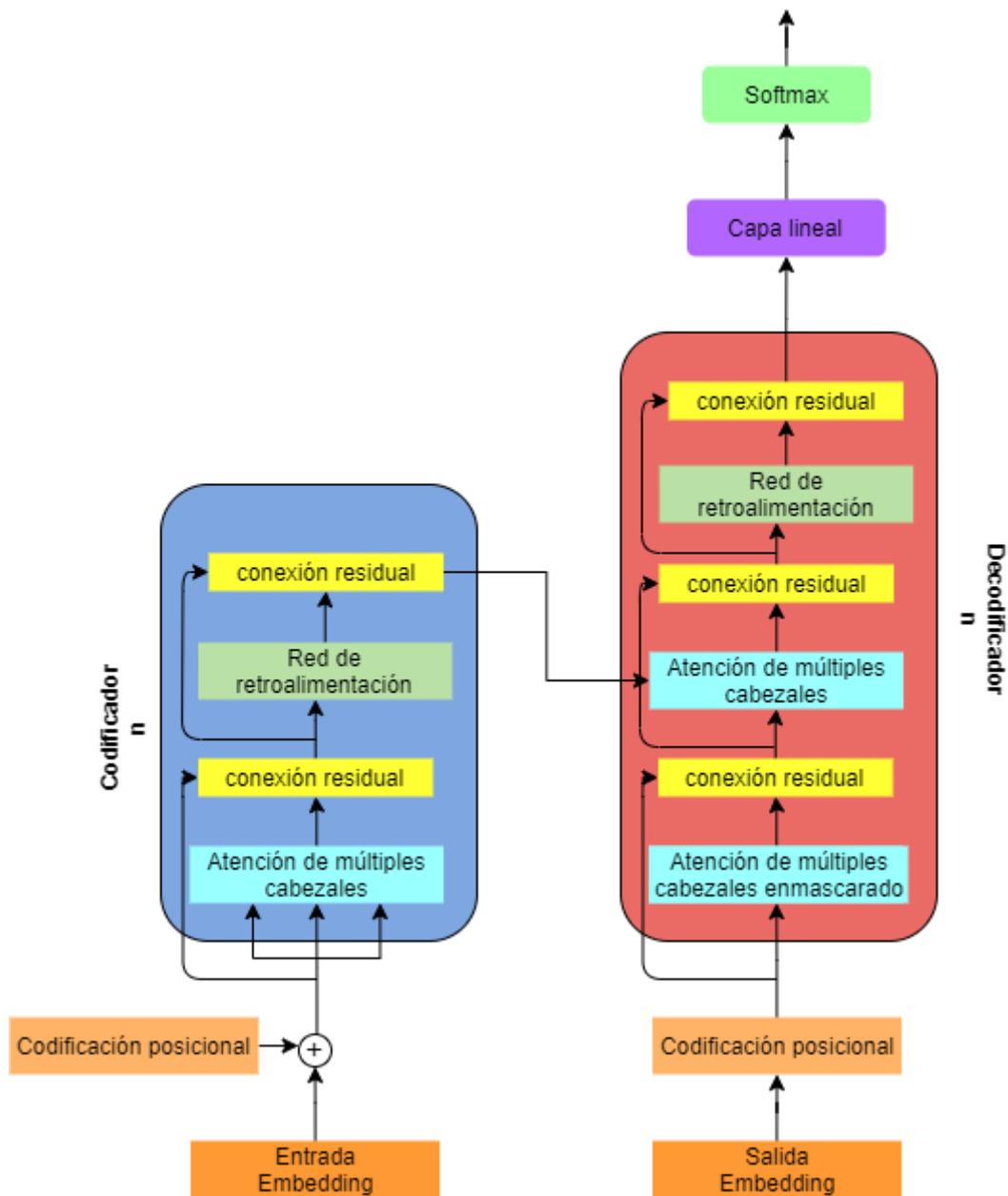


Figura 12: Estructura codificador-decodificador (Vaswani et al., 2017)

3.5 BERT

Publicado en el artículo intitulado BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding el año 2018 por un equipo de google BERT cuyas siglas significan *Bidirectional Encoder Representations from Transformers* (Representación de codificadores bidireccionales de Transformers en español) presenta ciertas ventajas en comparación de otros modelos como GPT en su versión 1 a la cual nos referiremos como GPT-1 y ELMo. Hablemos brevemente de estos modelos; ambos utilizan arquitectura *transformer*, sin embargo mientras GPT-1 utiliza una pila de decodificadores en los cuales se incluye un

enmascaramiento el cual oculta ciertas palabras de la red (asignándoles un valor de 0) con la finalidad de entrenar a la red (Radford et al., 2018); en el caso de BERT este no utiliza decodificadores si no codificadores a pesar de ello también realiza un enmascaramiento. la otra diferencia clave está en el entrenamiento de BERT que se compone de dos fases, la primera es el pre-entrenamiento donde aprende a entender que es el lenguaje y el contexto de las palabras, la segunda fase es llamada Fine-tuning (ajuste fino) fase en la que se le enseña a el modelo a realizar tareas en específico. Durante el pre-entrenamiento podemos ver diferencias entre los modelos por ejemplo GPT-1 visto en la figura 13⁹, sólo es capaz de visualizar *embeddings* de izquierda a derecha, esto es importante porque GPT-1 tratará de predecir la siguiente palabra dada la palabra anterior, si fuera bidireccional como lo es BERT el modelo simplemente haría trampa viendo las palabras que siguen o anteceden y por lo tanto el entrenamiento no sería efectivo.

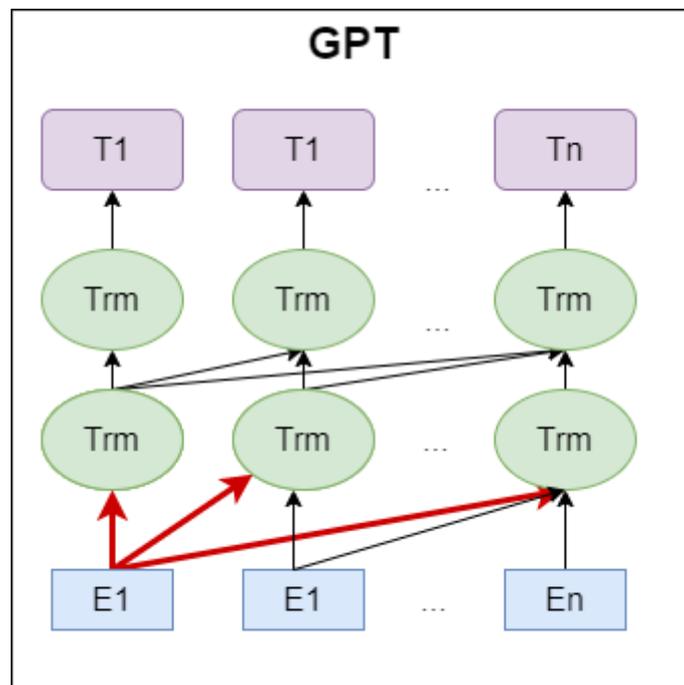


Figura 13: Arquitectura del pre-entrenamiento GPT-1 (Devlin et al., 2019)

Mientras que BERT es capaz de ir de derecha a izquierda y de izquierda a derecha, es por ello que es bidireccional tal como se muestra en la figura 14¹⁰

⁹ Inspirado de (Devlin et al., 2019)

¹⁰Inspirado de (Devlin et al., 2019)

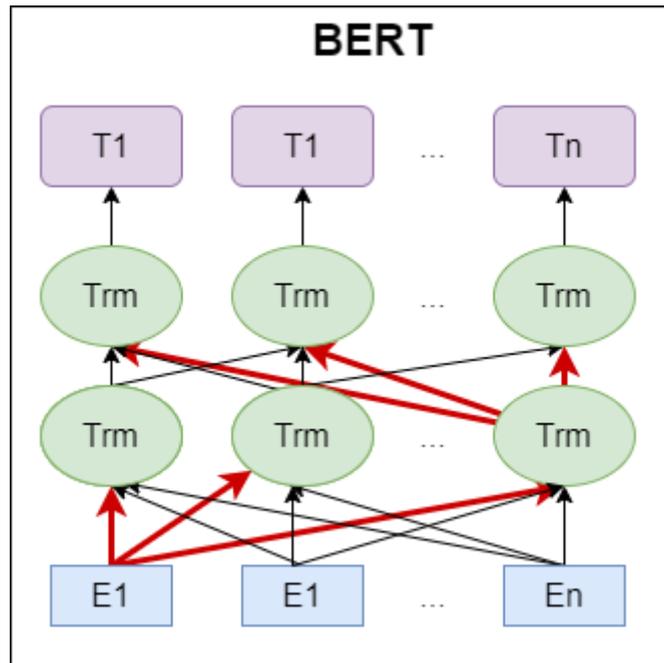


Figura 14: Arquitectura del pre-entrenamiento BERT (Devlin et al., 2019)

Como se mencionó con anterioridad, GPT-1 cuenta con decodificadores encargados de realizar el enmascaramiento en el pre-entrenamiento del modelo, si funcionaba de forma bidireccional no aprendería del lenguaje y entonces surge la siguiente pregunta ¿Porque BERT es bidireccional y aprende? Usemos un ejemplo para entender su funcionamiento, tenemos una oración sencilla *El perro juega con la pelota* eliminamos de forma aleatoria algunas palabras (en este caso se elimina la palabra juega) y se sustituye con un token de enmascaramiento quedando de la siguiente manera *El perro [mask] con la pelota* de esta forma tratará de predecir las palabras faltantes sin hacer trampa porque la palabra ya no está ahí.

BERT utilizó una gran cantidad de textos, así como artículos de la wikipedia para su pre-entrenamiento el cual se dividió en dos tareas la primera el modelado de lenguaje donde se ocultó el 15 % de las palabras que tenía que predecir y la segunda tarea fue la predicción de la siguiente oración, que en conjunto forman el pre-entrenamiento de BERT. Ahora veamos que es el Fine-tuning que consiste en ingresar un conjunto de datos etiquetados para poder realizar tareas en específico, siendo la fase de pre-entrenamiento la que implica un alto costo de recursos, mientras que realizar Fine-tuning resulta mucho más barato respecto a los recursos utilizados. (Devlin et al., 2019).

Además de estas bondades que existen en BERT, también es muy eficiente para las tareas de clasificación, es por ello que se decidió utilizar BERT en lugar de otros modelos ya mencionados.

Capítulo 4

Datos

Este capítulo presenta cuál fue el proceso para la obtención de los datos que se utilizaron para el entrenamiento del modelo, su procesamiento y su clasificación, cabe mencionar que algunos voluntarios ayudaron a la anonimización y el etiquetado de los datos.

4.1 Origen de los datos

Para entrenar un modelo de lenguaje, hay un ingrediente que no puede faltar, este ingrediente son los datos, los cuales son determinantes para que este pueda aprender a realizar diferentes tareas que se le enseñan, en nuestro caso la tarea es la clasificación de emociones. Estos conjuntos de datos son conocidos como corpus y se pueden encontrar de forma pública y gratuita en internet, debido a que existen muy pocos corpus para la clasificación de emociones en español se decidió recolectar datos a través de donaciones de diferentes personas, para ello se utilizó un formulario de google que se compartió a diferentes personas en el cual se pedía que de forma voluntaria y anónima compartieran un día de su vida de forma escrita al estilo de un diario, se puede consultar una copia del cuestionario dando clic [aquí](#).

4.2 Anonimización

Una vez se hizo la elicitación de corpus a lo largo de tres semanas, tiempo determinado para la obtención de los datos en el cual se obtuvo la participación de 121 personas que decidieron donar sus historias, para el entrenamiento de este modelo, fue importante que los datos proporcionados se mantengan anónimos por lo que se revisaron las historias donadas y quito la información sensible que pueda aparecer como:

- Direcciones
- Números de teléfono
- Contraseñas
- Nombres con apellido
- Correos electrónicos.

A continuación se muestran ejemplos de la forma en la que se deben abordar estas situaciones.

Supongamos que se tiene la siguiente historia:

La historia que quiero contar es sobre mi mamá, su nombre es Juana Meliza Martínez Anaya, tiene 44 años y vive conmigo en Av. Parque Lira S/N, Bosque de Chapultepec I

Secc, Miguel Hidalgo, 11850 Ciudad de México, CDMX . Ella es una persona muy especial para mí, somos mejores amigas. Sonará extraño, pero le cuento todo a mi mamá, hasta la contraseña de mi facebook, se rió cuando le dije que mi contraseña era micontraseña1234, me dijo que no hacía falta decírsela, que seguramente cualquiera la adivinaría. Es una mujer muy inteligente, quiero ser como ella, ella es doctora y trabaja en el Hospital Ángeles Lindavista de lunes a viernes en un horario de 5:00 am a 2:00 pm, estoy muy orgullosa de ella, yo también quiero ser doctora, espero algún día ser tan excelente como lo es ella. Algo que me gusta mucho de ella es que no solo trabaja en el hospital, sino que también visita a pacientes que necesitan de atención médica, les dejo su número por si algún día necesitan atención médica, es 5512345678. Como pueden notar, es una mujer excepcional, siempre apoya a los demás. La amo.

Después de haber leído la historia, notarás que aunque la historia es positiva y cargada aparentemente de buenas intenciones, hay datos que hacen cada vez más peligroso el hecho de que esta historia pueda ser leída por cualquier persona. Así que será necesario censurar ciertos datos que aquí se muestran, por lo que los pasos que se muestran a continuación fueron los que se siguieron para la anonimización de los datos:

Paso 1. Identificación de datos sensibles:

La historia que quiero contar es sobre mi mamá, su nombre es Juana Meliza Martínez Anaya, tiene 44 años y vive conmigo en Av. Parque Lira S/N, Bosque de Chapultepec I Secc, Miguel Hidalgo, 11850 Ciudad de México, CDMX . Ella es una persona muy especial para mí, somos mejores amigas. Sonará extraño, pero le cuento todo a mi mamá, hasta la contraseña de mi facebook, se rió cuando le dije que mi contraseña es micontraseña1234, me dijo que no hacía falta decírsela, que seguramente cualquiera la adivinaría. Es una mujer muy inteligente, quiero ser como ella, ella es doctora y trabaja en el Hospital Ángeles Lindavista de lunes a viernes en un horario de 5:00 am a 2:00 pm, estoy muy orgullosa de ella, yo también quiero ser doctora, espero algún día ser tan excelente como lo es ella. Algo que me gusta mucho de ella es que no solo trabaja en el hospital, sino que también visita a pacientes que necesitan de atención médica, les dejo su número por si algún día necesitan atención médica, es 5512345678. Como pueden notar, es una mujer excepcional, siempre apoya a los demás. La amo.

Paso 2. Censura y/o sustitución de información:

- Para sustituir el nombre, se usó una tabla nombres genéricos o comunes que aparecen en la tabla 1
- La dirección deberá ser censurada en su totalidad.
- La contraseña deberá ser censurada en su totalidad
- El lugar de trabajo debe ser censurado en su totalidad.
- El horario de trabajo debe ser censurado en su totalidad.
- El número telefónico debe ser censurado en su totalidad.

Masculinos	Femeninos	Neutros
Juan	María	Guadalupe
Francisco	Guadalupe	Robin
Luis	Ana	Francis
Jesús	Carmen	Denis
Carlos	Fernanda	Dani

Tabla 1: Nombres genéricos

Paso 3. Revisar la coherencia de la historia posterior a las modificaciones.

No solo se trató de eliminar información sino también se debió cuidar que la historia no perdiera coherencia siendo lo más cercana a lo que se quiere comunicar cuidando también de no eliminar palabras claves para el la identificación de las emociones.

Regresando al ejemplo se consideró las siguientes oraciones sensibles extraídas de la historia original:

- *...vive conmigo en Av. Parque Lira S/N, Bosque de Chapultepec I Secc, Miguel Hidalgo, 11850 Ciudad de México, CDMX*

Para esta oración lo adecuado sería eliminar la dirección completa, pero conservar la oración “vive conmigo”.

- *...Sonará extraño, pero le cuento todo a mi mamá, hasta la contraseña de mi facebook, se rió cuando le dije que mi contraseña es micontraseña1234*

Lo adecuado sería en este caso eliminar la contraseña y modificar oración para que tenga coherencia quedando de la siguiente forma “se rió cuando le dije mi contraseña”

- *...trabaja en el Hospital Ángeles Lindavista de lunes a viernes en un horario de 5:00 am a 2:00 pm*

Aquí lo correcto sería conservar la oración “trabaja en el Hospital”, y eliminar completamente el resto de la oración.

Paso 4. Resultado final.

Después analizar la historia y seguir los pasos anteriores, el resultado final adecuado es el siguiente:

La historia que quiero contar es sobre mi mamá, su nombre es María, tiene 44 años y vive conmigo. Ella es una persona muy especial para mí, somos mejores amigas. Sonará extraño, pero le cuento todo a mi mamá, hasta la contraseña de mi facebook, me dijo que no hacía falta decírsela, que seguramente cualquiera la adivinaría. Es una mujer muy inteligente, quiero ser como ella, ella es doctora y trabaja en el Hospital, estoy muy orgullosa de ella, yo también quiero ser doctora, espero algún día ser tan excelente como lo es ella. Algo que me gusta mucho de ella es que no solo trabaja en el hospital, sino que

también visita a pacientes que necesitan de atención médica. Como pueden notar, es una mujer excepcional, siempre apoya a los demás. La amo.

Se puede observar las diferencias entre la historia original y el resultado final en la cual no se ha modificado la parte importante de la historia ni se ha cambiado el sentido ni el mensaje que se desea transmitir. Este es el tipo de resultado al que se debe aspirar para un mejor modelo.

4.2 Etiquetado

Ya tenemos datos sin información sensible, lo siguiente para continuar con el procesamiento de los datos es etiquetar los mismos para poder entrenar al modelo, a continuación se presentan los criterios que se tomaron en cuenta para poder realizar el correcto etiquetado de los datos. Para ello se utilizó la plataforma de Doccano la cual se describe a continuación así como la clasificación de las emociones que se van a utilizar, las etiquetas y ejemplos que representen cada uno de las emociones listadas.

Doccano es una herramienta desarrollada por el IIMAS que nos permite realizar el etiquetado en un set de datos definido es por ello que será de utilidad para etiquetar los datos que se necesitan para este trabajo, el etiquetado se realiza de forma manual con cada una de las historias que se recolectaron, asignándoles la emoción que se identifica en dicho texto, el tipo de etiquetado que se utiliza es BIO donde al inicio de la emoción se le coloca una etiqueta B que significa begin es decir el comienzo de mi emoción, todo lo que se encuentra dentro de mi emoción es asignado con una etiqueta I que significa inside y aquello que no pertenece a alguna emoción se es asignado con una etiqueta O de outside es decir no aporta valor al etiquetado, si deseas obtener más información o utilizar la plataforma consulta la pagina <https://doccano.iimas.unam.mx/>.

Para este trabajo se han identificado seis sentimientos generales a los cuales se les asignó una etiqueta y un color.

- **Felicidad**
- **Enojo**
- **Asco**
- **Tristeza**
- **Sorpresa**
- **Miedo**

Estas son los 6 sentimientos base a identificar, los cuales son consecuencia de una serie de emociones cada vez más complejas como se muestra a continuación:

- La **Felicidad** es una sentimiento a la cual se llega mediante la transición de un estado emocional a otro, las emociones que participan en la transición para alcanzar la felicidad se muestran en la siguiente tabla:

Felicidad					
Alegre	Interesado	Orgullosa	Aceptado	Poderoso	Pacífico
Intimo	Optimista	Liberado	Eufórico	Entretenido	Curioso
Importante	Seguro	Respetado	Satisfecho	Valiente	Provocativo
Cariñoso	Esperanzado	Sensible	Bromista	Abierto	Inspirado

Tabla 2: Transición a la Felicidad

- El **Enojo** es una emoción a la cual se llega mediante la transición de un estado emocional a otro, las emociones que participan en la transición para alcanzar el enojo se muestran en la siguiente tabla:

Enojo					
Herido	Amenazado	LLeno de odio	Agresivo	Frustrado	Distante
Crítico	Apenado	Devastado	Atacado	Celoso	Resentido
Ultrajado	Furioso	Rabioso	Provocador	Hostil	Enfurecido
Irritado	Introvertido	Desconfiado	Escéptico	Sarcástico	Loco

Tabla 3: Transición a el enojo

- El **Asco** es una emoción a la cual se llega mediante la transición de un estado emocional a otro, las emociones que participan en la transición para alcanzar el asco se muestran en la siguiente tabla:

Asco					
Disconforme	Decepcionado	Horrible	Abstinencia	Moralista	Recio
Repugnante	Revoltoso	Asco	Odioso	Aversión	Vacilante

Tabla 4: Transición a el asco

- La **Tristeza** es una emoción a la cual se llega mediante la transición de un estado emocional a otro, las emociones que participan en la transición para alcanzar la tristeza se muestran en la siguiente tabla:

Tristeza					
Culpable	Abandonado	Desesperado	Deprimido	Solo	Aburrido
Arrepentido	Avergonzado	Ignorado	Victimizado	Desvalido	Vulnerable
Melancólico	Vacío	Desamparado	Aislado	Apático	Indiferente

Tabla 5: Transición a la tristeza

- La **Sorpres**a es una emoción al cual se llega mediante la transición de un estado emocional a otro, las emociones que participan en la transición para alcanzar la sorpresa se muestran en la siguiente tabla:

Sorpresa					
Sorprendido	Confundido	Asombrado	Entusiasmado	Conmocionado	Abatido
Desilusionado	Perplejo	Estupefacto	Impresionado	Entusiasta	Enérgico

Tabla 6: Transición a la sorpresa

- El **Miedo** es una emoción al cual se llega mediante la transición de un estado emocional a otro, las emociones que participan en la transición para alcanzar el miedo se muestran en la siguiente tabla:

Miedo					
Humillado	Rechazado	Sumiso	Inseguro	Ansioso	Asustado
Ridiculizado	Irrespetado	Alienado	Marginado	Insignificante	Inutil
Inferior	Insuficiente	Preocupado	Agobiado	Espantado	Aterrado

Tabla 7: Transición a el miedo

Una vez definidas las etiquetas podemos realizar el etiquetado de los datos de la siguiente forma

- 1.- Accedemos a la plataforma de doccano con nuestro usuario y contraseña a través de la siguiente liga <https://doccano.iimas.unam.mx/auth>
- 2.-En la esquina superior derecha accedes a Projects
- 3.- Seleccione el proyecto Chatbot_Sentimientos
- 4.- Seleccione en la columna de la izquierda la opción de Dataset
- 5.- Da clic donde dice “anótate” de alguna de las filas que elijas y listo.

A toda acción corresponde una reacción, es por ello que no solo se debe identificar la emoción si no la causa o motivo de esta, a continuación se mostraran los criterios que se tomaron en cuenta para la etiquetación de los datos, supongamos que tenemos el siguiente texto:

“Hoy me he levantado y me di cuenta que no sonó mi alarma por lo que me perdi de la mañanera de amlo y me siento triste”

En la historia podemos identificar que se trata de **tristeza** sin embargo no todo el texto es de utilidad por lo que debemos etiquetar el texto de la siguiente manera:

*“Hoy me he levantado y me di cuenta que no sonó mi alarma por lo que **me perdi de la mañanera de amlo y me siento triste**”*

Donde se marca de color **verde** la causa y de color **azul** el efecto, por lo que debemos de etiquetar tanto causa y el efecto con la etiqueta de tristeza quedando de la siguiente forma:

*“Hoy me he levantado y me di cuenta que no sonó mi alarma por lo que **me perdi de la mañanera de amlo y me siento triste**” **tristeza***

Existe la posibilidad que en el texto no esté ordenado de la forma causa efecto si no al revés efecto causa por ejemplo:

*“**Hoy me he levantado triste** **porque no sonó mi alarma y me perdi de la mañanera de amlo**”*

De igual forma se marca de color **verde** la causa y de color **azul** el efecto, por lo que debemos de etiquetar tanto causa como efecto con la etiqueta de tristeza quedando de la siguiente forma:

*“**Hoy me he levantado triste** **porque no sonó mi alarma y me perdí de la mañanera de amlo**” **tristeza***

Otro caso es en el que no se encuentre la palabra triste si no alguna de sus transiciones como se mostró en la tabla 5 de transiciones a la tristeza por ejemplo:

“Hoy desayune y me dispuse a ir a trabajar como todos los días pero cuando abrí la puerta de mi casa mi perra sasha escapó, la busque por 3 horas pero no la encontré por lo que me sentí abandonado”

Se trata de una historia donde está la tristeza pero no aparece la palabra triste si no **abandonado** la cual se encuentra en la tabla de transiciones de emociones hacia la tristeza por lo que el etiquetado queda de la siguiente manera:

*“Hoy desayune y me dispuse a ir a trabajar como todos los días pero cuando abrí la puerta de mi casa **mi perra sasha escapó, la busque por 3 horas pero no la encontré por lo que me sentí abandonado**”*

En una historia puede que se encuentren más de una emoción por ejemplo:

“El día de hoy me levante por la mañana un poco nostálgica de saber que la Navidad se acerca y mi esposo y yo no tenemos mucho presupuesto para la compra del árbol y la cena, así que prendí la cafetera, hice un poco de meditación por la mañana como me lo recomienda a diario mi psicoterapeuta y después di de comer a los gatos que se llaman Marco y Polo en honor al comerciante Italiano y viajero. Después comencé a leer un libro digital que se llama Mujercitas o The Little Women poco a poco me fui identificando con ellas pues vivían un momento fuerte en sus vidas y la Navidad no fue un festejo o celebración total, en ese tiempo vivían los Americanos la Guerra Civil, y si bien yo no vivo una guerra ¡Gracias a Dios! si vivo como todos la pandemia del Covid T19, después dejé el libro entreabierto en esa página y me fui muy contenta a decirle a mi esposo que aunque la Navidad no sea esta vez con grandes regalos, mi único presente es tener a mi familia con salud y a los gatitos que alegran mis días , así me sentí mucho mejor ” **Tristeza Miedo Felicidad**

En esta historia existen 3 sentimientos **Tristeza Miedo Felicidad** además “me sentí mucho mejor” no tiene causa y tampoco viene la ninguna de las emociones que están en la tabla de la felicidad así como “mi único presente es tener a mi familia con salud y a los gatitos que alegran mis días” o “¡Gracias a Dios!” sin embargo se identifica como una transición hacia la **Felicidad** por lo que deben etiquetarse como felicidad, se tiene que tener cuidado de no dejar pasar esos detalles ya que son importantes para el correcto etiquetado y entrenamiento del modelo.

Tabla 8 ejemplo de expresiones que denotan emociones que no se encuentran en las tablas anteriores:

Expresión	Emoción
jajajaja	Felicidad
Gracias a dios	Felicidad
¡por fin vacaciones!	Felicidad

Tabla 8 : Expresiones que denotan alguna emoción

4.3 Stop Words

En este estado los datos ya se encuentran libres de información sensible y etiquetados, pero todavía queda un último paso el cual es quitar las palabras vacías o mejor conocidas como *stop words*, las cuales son palabras que no aportan nada al entrenamiento del modelo, por el contrario hacen que el proceso de entrenamiento tome más tiempo debido a que se utiliza una mayor cantidad de tokens, es por esta razón que se deben eliminar estas palabras de los datos, para ello se creó un script que realiza lo siguiente:

1. Recibe un archivo de texto que contiene los datos de entrenamiento del modelo y se agregan a una lista
2. Recibe un archivo que contiene las *stop words* utilizadas en el idioma español, estas palabras las cuales principalmente son preposiciones, conjunciones, artículos y pronombres por ejemplo, “a”, “el”, “la”, “lo”, “que” entre otras y se agregan a una lista
3. Se itera la lista que contiene los los datos en caso de que encuentre alguna palabra que se encuentre en la lista de palabras vacías esta se elimina de la lista de datos y de esta forma se obtiene una lista libre de *stop words*

Para ello se utilizó lenguaje de programación python en un archivo de google colab que se puede consultar en el [enlace](#) donde viene el proceso que se realizó.

4.4 Análisis de los datos

El [script](#) desarrollado también realiza un filtrado de los datos donde se realizaron 3 transformaciones a los datos, en la primera las letras mayúsculas son sustituidas por minúsculas, en la segunda se quitan las *stop words* a partir de los datos originales y en la última de igual forma se quitan las *stop words* pero a partir de los datos que ya fueron cambiados a minúsculas. Para mostrar las diferencias que existen al realizar estas transformaciones respecto a los datos sin modificar, en cada estado de los datos se realizó una tabla con las 20 palabras que más se repiten, una gráfica de barras con las 25 palabras más frecuentes y por último un word cloud con las 50 palabras que más aparecen, A continuación se muestra el análisis de los datos obtenidos.

20 Palabras que más se repiten							
de	572	me	357	no	186	lo	103
que	554	mi	277	con	180	día	100
y	535	en	262	para	146	pero	96
a	513	el	216	por	129	al	91
la	411	un	201	las	122	se	90

Tabla 9: lista de las 20 palabras que más se repiten

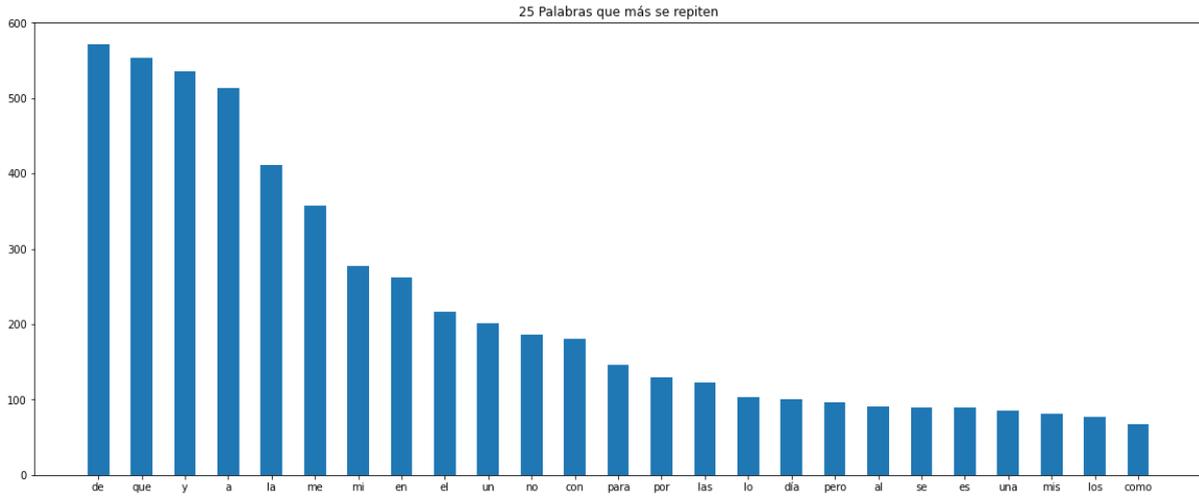


Figura 15: Gráfico de las 25 palabras que más se repiten



Figura 16: Word cloud de las 50 palabras que más se repiten

20 Palabras que más se repiten en minúsculas							
de	577	me	378	no	191	al	109
que	554	mi	290	con	180	lo	108
y	539	en	271	para	150	día	100
a	523	el	238	por	139	pero	100
la	421	un	204	las	124	se	90

Tabla 10: Lista de las 20 palabras que más se repiten en minúsculas

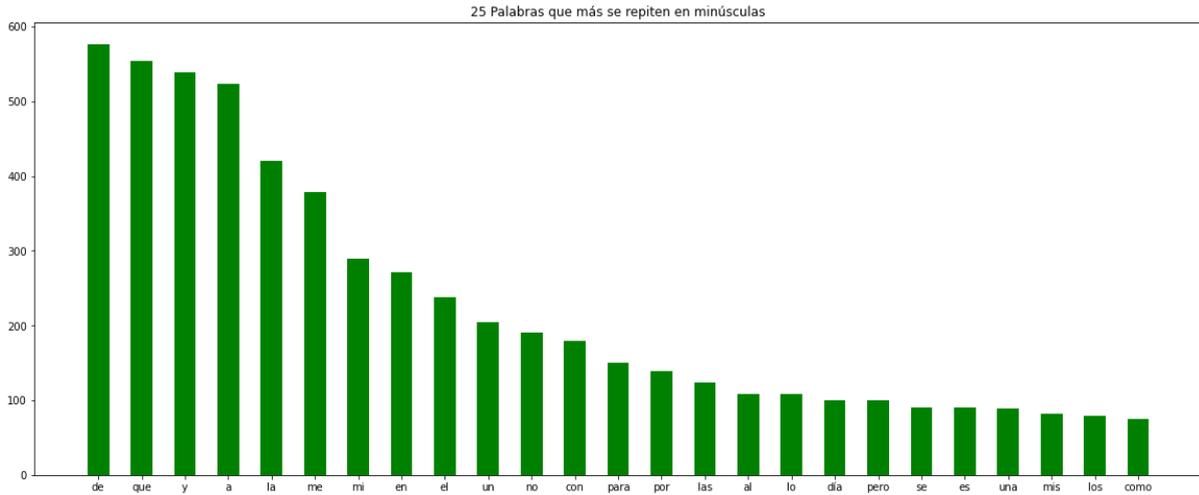


Figura 17: Gráfico de las 25 palabras que más se repiten en minúsculas



Figura 18: Word cloud de las 50 palabras que más se repiten en minúsculas

20 Palabras que más se repiten sin stop words							
día	100	vez	29	casa	25	El	22
después	45	había	29	Después	25	comer	22
ver	42	puse	29	tenía	25	Me	21
le	40	hora	26	tarde	23	está	21
Hoy	31	hoy	26	son	23	mamá	21

Tabla 11: Lista de las 20 palabras que más se repiten sin *stop words*

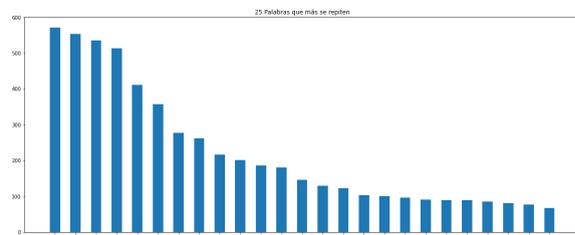


Figura 21: Gráfico de las 25 palabras normalizadas que más se repiten sin *stop words*



Figura 22: Word cloud de las 50 palabras normalizadas que más se repiten sin *stop words*

Se cuenta con un total de **3616** palabras diferentes las cuales se reducen a **3468** a partir de la la primera transformación donde se convierten las letras mayúsculas a minúsculas, esto porque el lenguaje de programación distingue entre mayúsculas y minúsculas, viéndose reducido en un **4.09%** el total de palabras, realizando la segunda transformación es decir quitar las



stop words tenemos un total de **3407** palabras diferentes lo que equivale a una reducción del **5.77%** y en la última transformación donde se quitaron las stop words y las letras mayúsculas se convirtieron en minúsculas se cuenta con un total de **3258** palabras lo equivalente a una reducción del **9.9%** de palabras diferentes. Todo esto es importante porque recordando el capítulo 3 y el principio de este capítulo cada palabra es equivalente a un token por lo tanto reducir la cantidad de tokens es reducir los recursos y tiempo necesario para entrenar al modelo.



Capítulo 5

Pruebas y Resultados

En los capítulos anteriores el autor explicó sobre los chatbots, emociones, modelos de lenguaje, arquitectura del *transformer* y de los datos que se utilizaran para el desarrollo de este trabajo; a continuación se mostraran las pruebas que se realizaron para el entrenamiento del modelo y los resultados obtenidos.

Una vez ya contamos con los datos para el entrenamiento del modelo es necesario desarrollar código para que la información sea procesada correctamente y obtengamos los resultados que esperamos, para ello Hugging Face comparte toda la información necesaria para poder entrenar un modelo ya sea desde su [sitio web](#) o su repositorio de [GitHub](#). Para este trabajo se tomó como referencia un notebook intitulado “02-transformers.ipynb” del repositorio de GitHub de Hugging Face, al cual se le realizaron modificaciones para cumplir con el fin de este trabajo, el notebook se puede encontrar en la siguiente [liga](#).

Para el entrenar al modelo se utilizaron dos modelos de lenguaje en idioma español, el primero es “bert-base-spanish-wwm-cased” desarrollado por el departamento de ciencias de la computación de la Universidad de Chile, el segundo el cual es más pequeño es “electra-small-spanish” desarrollado por la compañía Skim AI Technologies. A cada uno de estos modelos se les realizaron pruebas con un tamaño de batch de 16 referente al número de ejemplos de entrenamiento utilizados en cada iteración y una variación en la cantidad de épocas comenzando en 10 épocas, 25 épocas, 50 épocas y terminando en 100 épocas para cada modelo de lenguaje utilizado; de igual manera para cada uno de los modelos se utilizó el siguiente texto el cual contiene ciertas faltas de ortografía “el día de hoy me siento bien porque saqué 10 en mi examen de historia, fui el que tuvo la calificación más alta y mis compañeros me felicitaron” para probar la capacidad de detectar emociones en ese texto por último se probará con diferentes textos al modelo que tuvo un mejor desempeño. A continuación se muestran los resultados de dichas pruebas.

Para poder medir los resultados se utilizó la librería *segeval* de python la cual proporciona ciertas métricas que nos ayudan a conocer el rendimiento del modelo por cada época que recibe de entrenamiento, la métrica que nos interesa conocer y comparar para la realización de las pruebas es la precisión del modelo referente al número de predicciones correctas dividido por el número total de predicciones, este resultados se multiplica por 100 para obtener un porcentaje, de esta forma podremos conocer qué tan bueno es el modelo, sin

embargo es necesario configurar al modelo para que entre en modo evaluación de esta forma se obtienen resultados más confiables pues se utilizan datos que no se habían visto durante el entrenamiento, a estos datos se les conoce como datos de prueba los cuales corresponden a un pequeño porcentaje del total de datos los cuales son completamente nuevos para el modelo. A continuación se muestran los resultados obtenidos.

La primera prueba fue entrenar con 10 épocas con BERT y electra como se muestra en la figura 24 observando una mayor precisión del modelo utilizando BERT con un 98.3% de precisión mientras que electra tiene una precisión de 75.6%, sin embargo al momento de evaluar los modelos electra obtiene un 72% de precisión mientras que BERT 71.9% casi lo mismo.

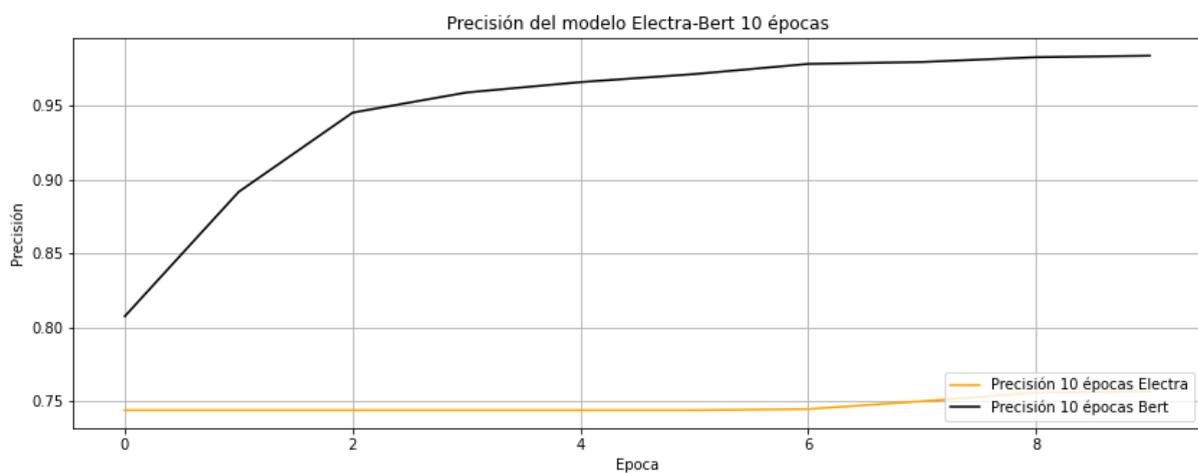


Figura 24: Precisión de BERT y electra 10 épocas

En la siguiente prueba se aumentó a 25 el número de épocas de entrenamiento, en la figura 25 se muestran los resultados, los cuales son favorables cuando se utiliza BERT como modelo de lenguaje con un 98.8% de precisión mientras que electra obtiene un 80.1% de precisión, cuando se evalúa la precisión del modelo esta es mayor cuando se usa BERT obteniendo un 73% comparado con el 70.1% de electra, contrario a lo ocurrido con 10 épocas de entrenamiento.

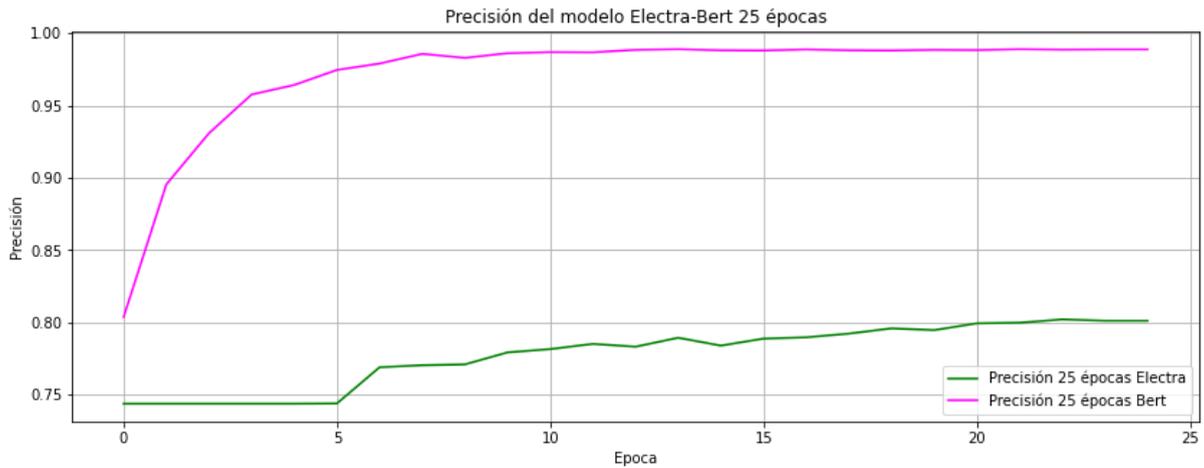


Figura 25: Precisión de BERT y electra 25 épocas

Continuamos aumentando el número de épocas de entrenamiento a 50 como se muestra en la figura 26, donde nuevamente al utilizar BERT se obtiene un mejor resultado 99.1% en comparación del 84.9% de electra, cuando los modelos se evalúan BERT logra un 72.2% y electra un 69.3% de precisión al momento de evaluarse.

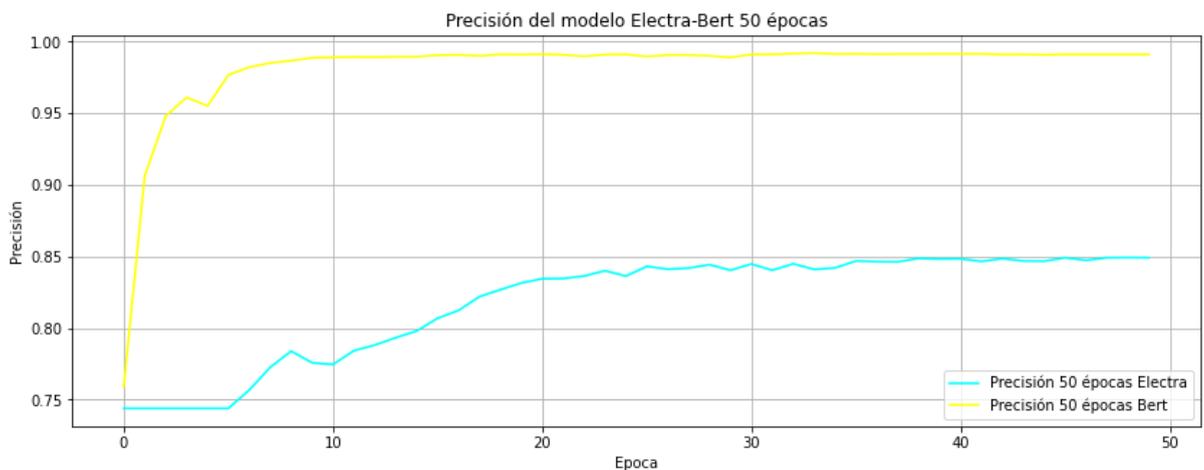


Figura 26: Precisión de BERT y electra 50 épocas

Por último se realizó una prueba con 100 épocas de entrenamiento, en la figura 27 se muestran los resultados obtenidos, en los cuales ambos modelos obtienen su mayor precisión BERT con 99.3% y electra con un 86.3% de precisión, cuando se evalúan los modelos electra obtiene un 69.1% de precisión contra un 72.8% de BERT a pesar de que esta no es el mejor resultado que obtiene al evaluar a BERT ya que con 25 épocas de entrenamiento su porcentaje de precisión es 0.2% mayor, al momento de evaluar al modelo con el texto que se sugirió al principio de este capítulo, BERT es la mejor opción de modelo de lenguaje para el propósito de este trabajo.

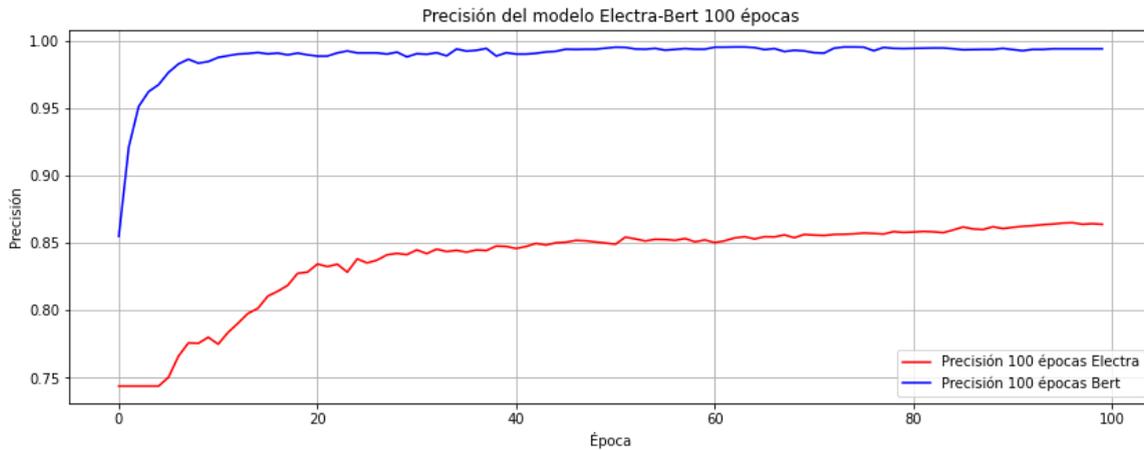


Figura 27: Precisión de BERT y electra 100 épocas

A continuación se muestra en la figura 28 los resultados de las pruebas que se realizaron al modelo BERT, en la figura 29 se muestran los resultados de electra, en la figura 30 se muestran los resultados de ambos modelos donde se observa de mejor manera el contraste que existe entre ellos y por último en la tabla 13 se muestran los resultados de la evaluación de ambos modelos.

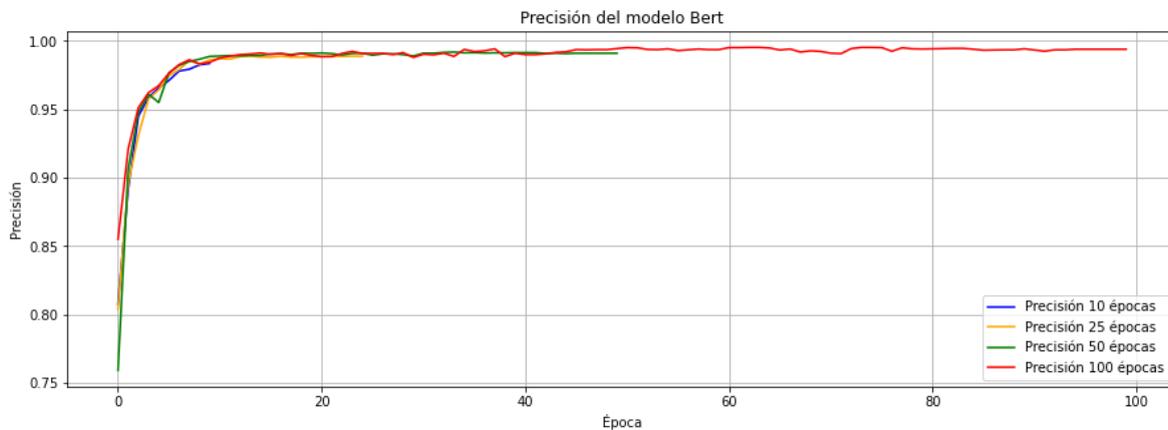


Figura 28: Precisión de BERT

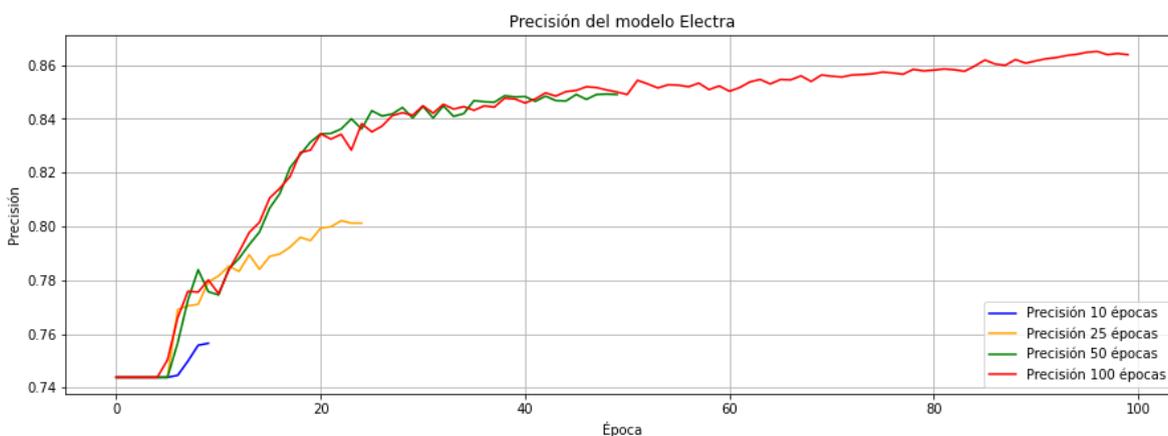


Figura 29: Precisión de electra

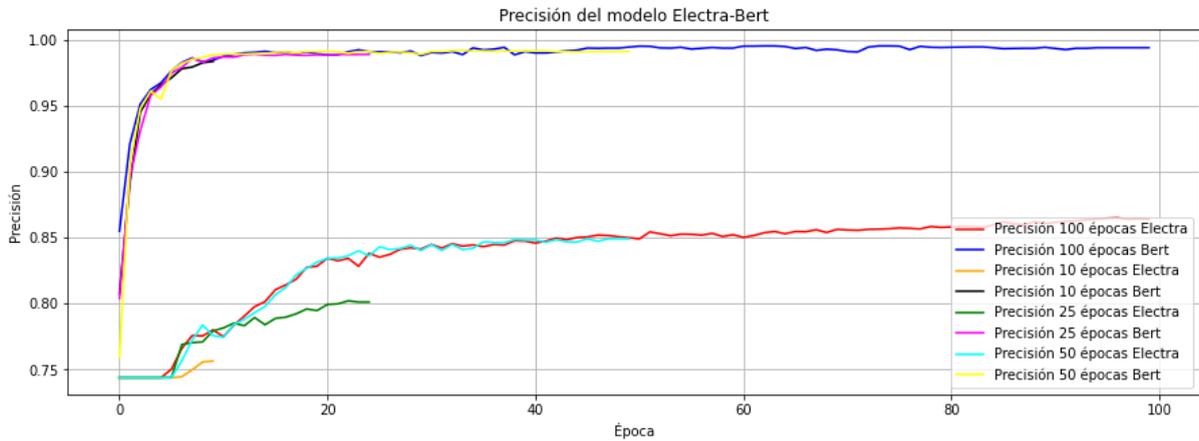


Figura 30: Precisión de BERT y electra

Modelo	10 épocas	25 épocas	50 épocas	100 épocas
% precisión BERT	0.983604158	0.98887425	0.991070122	0.993997951
% precisión electra	0.756528302	0.801207547	0.849056604	0.863849057

Tabla 13: Evaluación de BERT y electra diferentes épocas

Como resultado de estas pruebas se utilizará el modelo de lenguaje BERT para la realización de las siguientes pruebas del modelo que consistieron en analizar la precisión que se obtiene de cada sentimiento conforme cambia el número de épocas de entrenamiento. A continuación se muestran los resultados.

El primer sentimiento es el asco como se muestra en la figura 31 donde se observa una precisión del 100% utilizando 100 épocas de entrenamiento

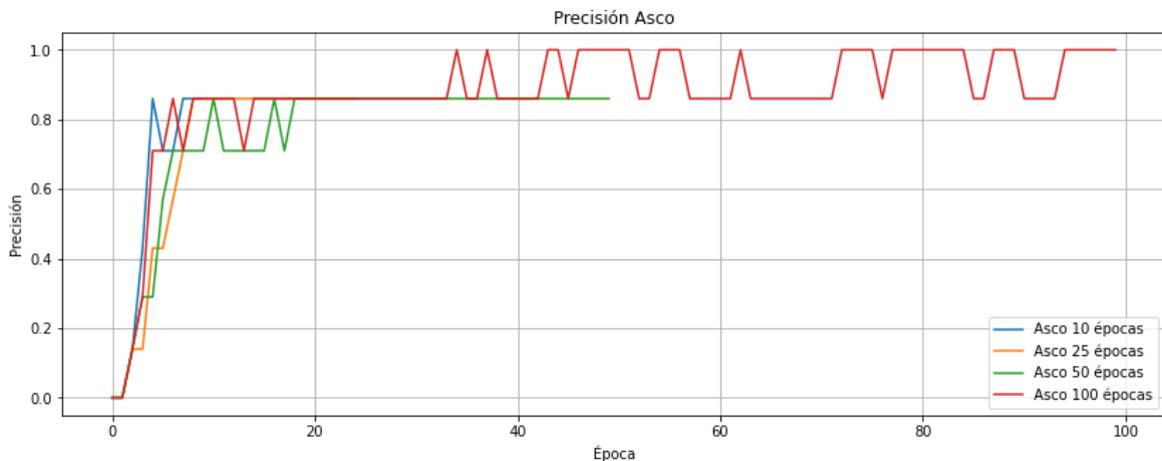


Figura 31: evolución de la precisión del asco en las diferentes épocas de entrenamiento

En el caso del enojo se obtiene una precisión del 89% con 25 y 50 épocas como se observa en la figura 32.

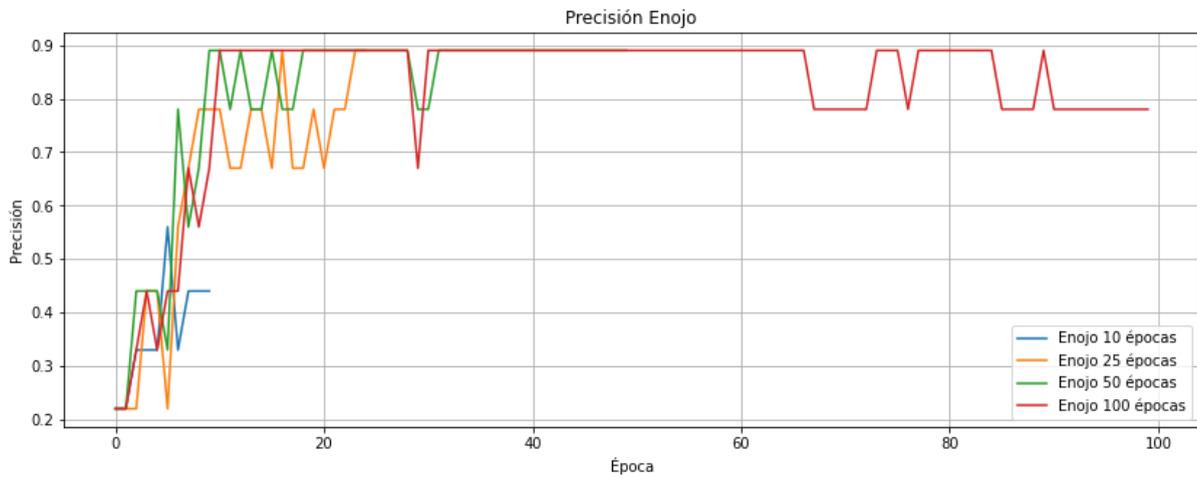


Figura 32: evolución de la precisión del enojo en las diferentes épocas de entrenamiento

La felicidad obtiene una precisión del 92% con un entrenamiento de 100 épocas mostrandose en la figura 33.

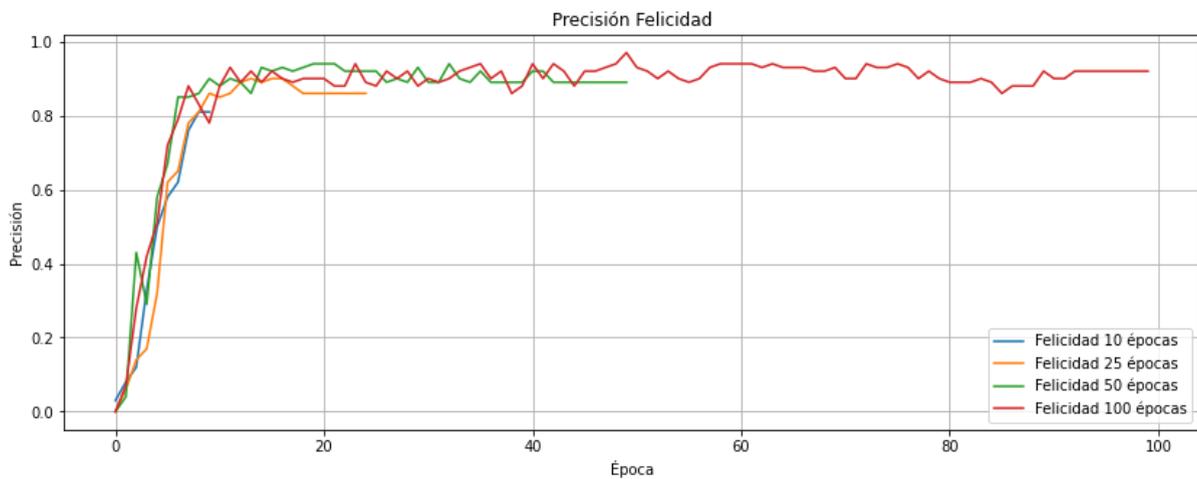


Figura 33: evolución de la precisión de la felicidad en las diferentes épocas de entrenamiento

En el caso del miedo se obtiene una precisión del 89% con 25, 50 y 100 épocas como se observa en la figura 34.

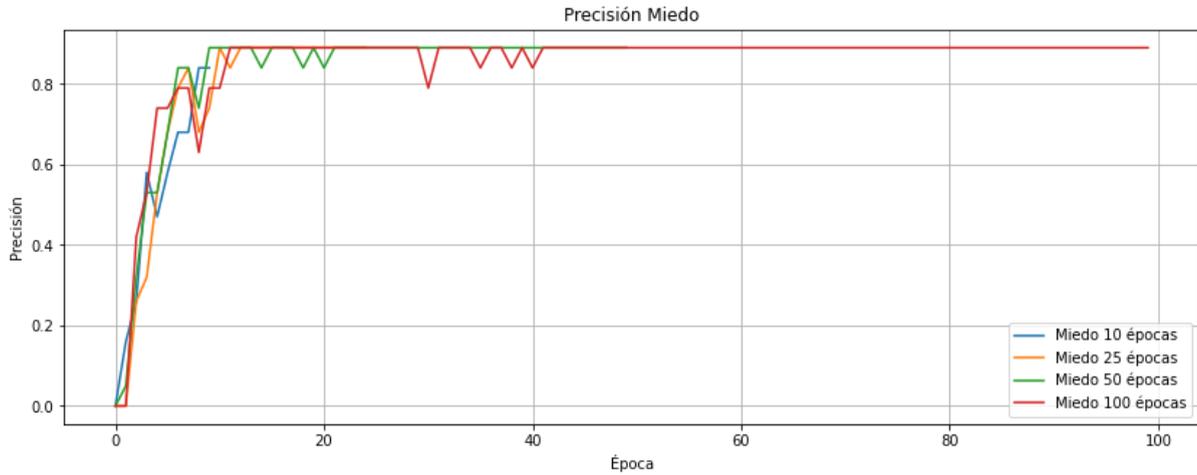


Figura 34: evolución de la precisión del miedo en las diferentes épocas de entrenamiento

En el caso de la sorpresa sucede que se obtiene la misma precisión de 83% entrenando 10, 25, 50 y 100 épocas como se observa en la figura 35.

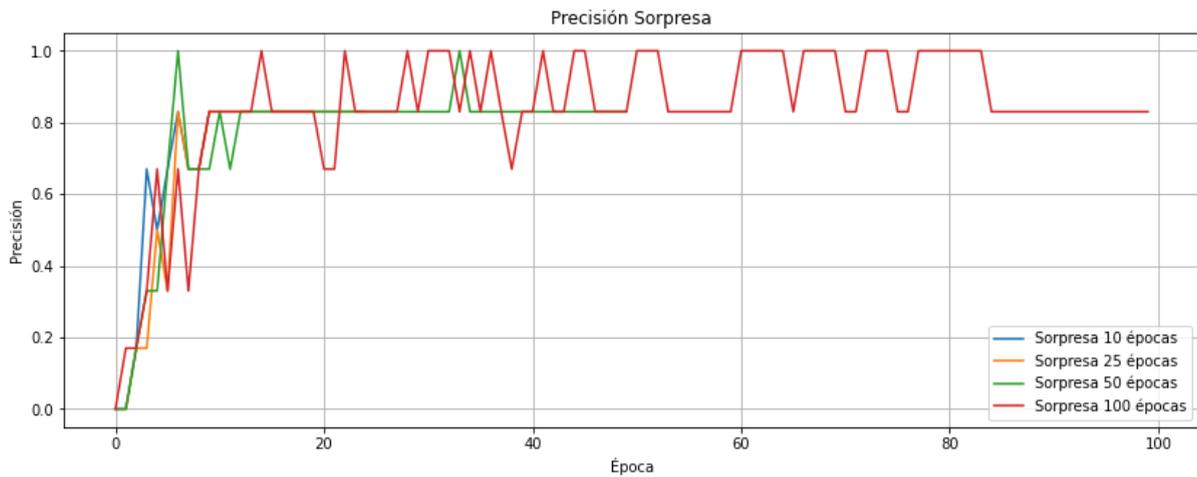


Figura 35: evolución de la precisión de la sorpresa en las diferentes épocas de entrenamiento

Por último en el caso de la tristeza la mayor precisión del 95% se obtiene entrenando 100 épocas como se observa en la figura 36.

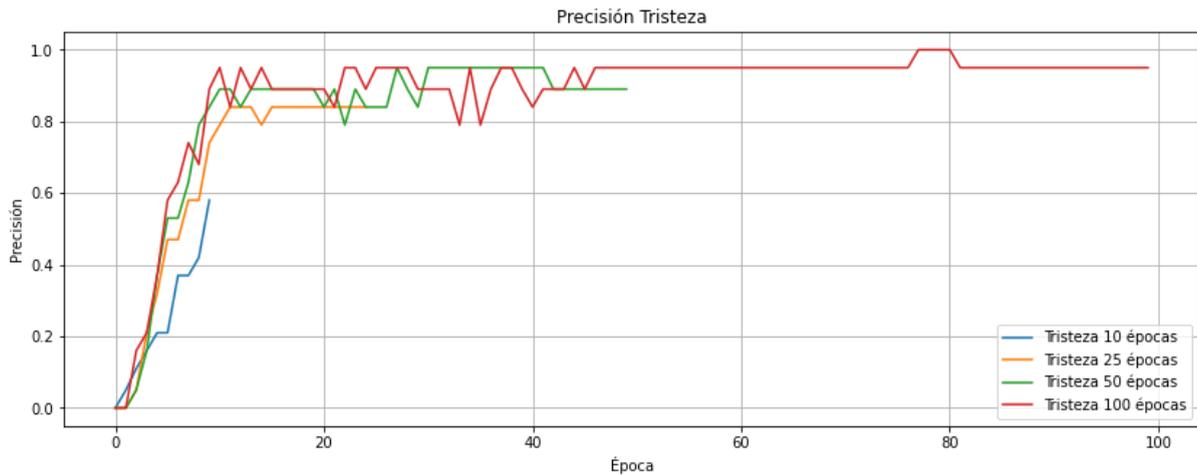


Figura 36: evolución de la precisión de la tristeza en las diferentes épocas de entrenamiento

Para finalizar las pruebas se decidió evaluar el desempeño del modelo con el texto que se propuso al inicio el cual está lleno de mucha carga positiva por lo que el modelo identifico el sentimiento de felicidad de la siguiente forma con 10, 25 50 y 100 épocas el cual se resalta con color amarillo, a continuación se muestran los resultados

- 10 épocas “el dia de hoy me siento bien porque saque 10 en mi examen de historia, fui el que tuvo la calificación mas alta y mis compañeros me felicitaron”
- 25 épocas “el dia de hoy me siento bien porque saque 10 en mi examen de historia, fui el que tuvo la calificación mas alta y mis compañeros me felicitaron”
- 50 épocas “el dia de hoy me siento bien porque saque 10 en mi examen de historia, fui el que tuvo la calificación mas alta y mis compañeros me felicitaron”
- 100 épocas “el dia de hoy me siento bien porque saque 10 en mi examen de historia, fui el que tuvo la calificación mas alta y mis compañeros me felicitaron”

El resultado que arroja el modelo es la etiquetación de cada palabra representada por un sentimiento en este caso el color amarillo representa la felicidad, sin embargo existen palabras que no pertenecen a ninguno de los sentimientos definidos por lo cual no se resaltaron, se observa que entre 10 y 25 épocas la diferencia en la identificación de las emociones es imprecisa, en cambio no existe diferencia cuando se usa el modelo de 50 y 100 épocas respecto al mismo texto, el resultado concuerda con lo que se mostró en la figura 33 en la cual al entrenar con 100 épocas se obtuvieron mejores resultados, por lo que

decidimos evaluar diferentes textos, cada uno con una carga emocional diferente utilizando el modelo de 100 épocas y estos fueron los resultados.

Cada emoción es representada por un color **Felicidad** **Enojo** **Asco** **Tristeza** **Sorpresa** **Miedo**.

- “Hoy fue un día malo, **perdi mi autobus en la mañana** por lo que tuve que irme caminando y llegue tarde a mi primer clase”
- “Me levante temprano por la mañana porque **un olor horrible me despertó**, **vi que era mi perro que hizo del baño** en la alfombra”
- “**estoy furiosa porque el señor de la tienda me dio un billete falso**”
- “En la tarde recibí una mala **noticia**, el gato de mi prima había **muerto**”
- “hoy **casi choco camino a mi casa**, **un camion me cerro el paso**, **mi mama grito del miedo**”

Se observa que los resultados que arroja el modelo no son del todo precisos, por lo que el modelo necesita ser trabajado aún más para lograr una mejor identificación de las emociones.

Para probar los diferentes modelos ejecuta el notebook correspondiente al número de épocas el siguiente [enlace](#).

Capítulo 6

Conclusiones

El trabajo desarrollado termina en este capítulo el cual aborda los aspectos más importantes que se vieron, se analizan las metas y objetivos que se propusieron con el fin de ver si estos se cumplieron o no y cuál es el trabajo a futuro.

En un inicio el trabajo surgió como una propuesta para ampliar las capacidades de un chatbot que estaba desarrollándose en el IIMAS lugar en el cual realicé mi servicio social así como algunas actividades relacionadas con el procesamiento de lenguaje natural, se plantearon diversas metas para lograr alcanzar el objetivo de entrenar un modelo de lenguaje el cual fuese entrenado utilizando la arquitectura *transformer*, se buscó que este pudiera identificar el flujo emocional que existe en las vivencias de las historias que se comparten a través de un chatbot llamado Wattson y para lograrlo se definieron diversas metas que ayudarán al cumplimiento del objetivo, como lo son la obtención de datos para entrenar al modelo, la anonimización de estas historias, el etiquetado de los datos y el entrenamiento del modelo. A lo largo de este escrito se describió el trabajo realizado para lograr estas metas, como lo es la creación de un formulario de google que se compartió a diferentes personas para que compartieran un día de su vida de forma escrita al estilo de un diario, con el cual se obtuvo un corpus lo suficientemente grande para entrenar al modelo a esta tarea se le conoce como elicitación de corpus, posterior a esto se procedió a la anonimización de los datos tarea para la cual se creó una guía que ayudó a su realización con la ayuda de voluntarios del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) que ayudaron a este procedimiento, una vez se anonimizaron los datos estos fueron etiquetados con la ayuda de los mismos voluntarios, tarea para la cual también se creó una guía para su correcta realización, posterior a la etiquetación se eliminaron las *stop words* y se realizó un análisis estadístico de los datos, para ello fue necesario la creación de dos scripts que facilitaron este trabajo. Posterior a la fase de obtención, anonimización y etiquetación de los datos, se entrenó el modelo, para ello se modificó un notebook obtenido del repositorio de github de hugging face, el cual fue modificado para que cumpliera con el propósito de este trabajo.

Se realizaron diferentes pruebas en las que se variaron los parámetros de entrenamiento como lo fue el del número de épocas de entrenamiento que se aplicaban al modelo y se probaron dos modelos de lenguaje, durante estas pruebas se evaluó la precisión de este, los resultados indicaron que la cantidad de épocas a utilizar para el entrenamiento del

modelo fueron 100 y el modelo de lenguaje a utilizar es *BERT-base-spanish-wwm-cased* con los cuales se obtuvieron mejores resultados al momento de poner al modelo en modo evaluación, los gráficos que se realizaron después de cada época de entrenamiento muestran que algunas emociones llegaban a tener una oscilación respecto a la precisión y el número de épocas, esto debido al algoritmo de optimización adam el cual se utilizó en el entrenamiento, el cual realiza estos saltos al tratar de encontrar un error mínimo global, también se observó que algunas emociones llegaron a tener una mayor precisión que otras e incluso alcanzaron su mejor rendimiento en épocas inferiores, esto debido a que la cantidad de datos etiquetados respecto a las categorías no está equilibrado, para obtener resultados más homogéneos se debió equilibrar el número de etiquetas, después del análisis de cuáles eran los mejores parámetros para el modelo, se puso a prueba con seis diferentes textos de ejemplo en los cuales identificó la carga emocional que existe, sin embargo en algunos de ellos el resultado no fue el esperado debido a que no logró identificar correctamente las emociones, esto posiblemente debido a que el tamaño del corpus utilizado tiene que ser aún más grande para que el modelo pueda aprender de una mayor cantidad de ejemplos.

A través de las metas y los resultados obtenidos se logró entrenar un modelo que puede identificar el flujo emocional de esta forma el objetivo del presente trabajo se logró cumplir, durante el desarrollo se identificaron áreas de oportunidad para el desarrollo de trabajos futuros algunas de las mejoras que se identificaron es la obtención de un corpus mucho más grande mediante la colaboración de personas que deseen compartir sus experiencias cotidianas, para ello una mayor difusión de la encuesta podría ser un método más eficiente para la obtención de una mayor cantidad de datos todo esto con la finalidad de mejorar la precisión del modelo, la implementación de este modelo al chatbot watson o a cualquier otro agente conversacional que pueda ser utilizado con fines educativos o de investigación, gracias a esto podría ser capaz de mediante el mismo método la adición de sentimientos aún más complejos, la etiquetación de textos mucho más grandes, todo esto se puede desarrollar y se puede tomar como punto de partida o referencia esta tesis.

Fuentes de información

- AbuShawar, B., & Atwell, E. (2015). ALICE Chatbot: Trials and Outputs. *Computación y Sistemas*, 19(4). <https://doi.org/10.13053/cys-19-4-2326>
- Agencia Digital de Innovación Pública. (2020). *Lanzan Gobierno De Ciudad De México Y Facebook Chatbot "Victoria" a WhatAapp para informar medidas sobre COVID-19.* COVID-19. <https://covid19.cdmx.gob.mx/comunicacion/nota/lanzan-gobierno-de-ciudad-de-mexico-y-facebook-chatbot-victoria-whataapp-para-informar-medidas-sobre-covid-19>
- Artificial Linguistic Internet Computer Entity. (2020). En *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Artificial_Linguistic_Internet_Computer_Entity&oldid=993396811
- Asterisk. (2014). *TTS y ASR - Asterisk Wiki*. https://www.wikiasterisk.com/index.php?title=TTS_y_ASR
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. <http://arxiv.org/abs/1810.04805>
- Ekman, P., & Oster, H. (1981). Expresiones faciales de la emoción. *Estudios de Psicología*, 2(7), 115–144. <https://doi.org/10.1080/02109395.1981.10821273>
- ELIZA. (2021). En *Wikipedia, la enciclopedia libre*. <https://es.wikipedia.org/w/index.php?title=ELIZA&oldid=132201371>
- Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., & Verplank, W. (1992). *ACM SIGCHI Curricula for Human-Computer Interaction*. Association for Computing Machinery. <https://doi.org/10.1145/2594128>
- IBM. (2021, abril 29). *Watson Assistant—Un chatbot que va un paso más allá, Watson Assistant*. <https://www.ibm.com/mx-es/cloud/watson-assistant>
- Mehlig, B. (2021). *Machine learning with neural networks*. <https://doi.org/10.1017/9781108860604>

Olivas, O. (2016). SimSimi. En *Wikipedia, la enciclopedia libre*.

<https://www.merca20.com/que-es-simsimi-y-por-que-esta-causando-euforia-en-redes-sociales/>

orientacionandujar. (2018, octubre 10). *Manipulativo espectacular Emociones básicas y una rueda de palabras emocionales*. Orientación Andújar - Recursos Educativos.

<https://www.orientacionandujar.es/2018/10/10/manipulativo-espectacular-emociones-basicas-y-una-rueda-de-palabras-emocionales/>

PARRY. (2021). En *Wikipedia*.

<https://en.wikipedia.org/w/index.php?title=PARRY&oldid=998275442>

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training*. 12.

UNAM. (2020). *Diseñan universitarios chat inteligente para afianzar relaciones comerciales*.

https://www.dgcs.unam.mx/boletin/bdboletin/2020_1032.html

Weizenbaum, J. (1976). *Computer power and human reason: From judgment to calculation*. Freeman.

Woebot Health. (2021). *Mental Health Chatbot*. Woebot Health. <https://woebothealth.com/>

Wysa. (2021). *Wysa—Your safe space in this difficult time*. Wysa - Your Safe Space in This Difficult Time. <https://www.wysa.io/>