**UNIVERSIDAD MAYOR REAL Y PONTIFICIA DE SAN FRANCICO XAVIER DE CHUQUISACA**



**GENERADOR DE POEMAS UTILZANDO TRANSFORMS APLICANDO LA TECNICA DE FINE TUNING**

**UNIVERITARIO**: Juan Sergio Pita Colque

**CARRERA**: ING. EN CIENCIAS DE LA COMPUTACIÓN

**MATERIA**: DESARROLLO DE APLICACIONES INTELIGENTES (SIS-330)

SUCRE - BOLIVIA

Contenido

[1. RESUMEN 3](#_Toc76743417)

[2. ANTECEDENTES 3](#_Toc76743418)

[3. PROBLEMÁTICA 3](#_Toc76743419)

[4. PROPUESTA 3](#_Toc76743420)

[5. OBJETIVO GENERAL 3](#_Toc76743421)

[6. OBJETIVOS ESPECÍFICOS 4](#_Toc76743422)

[7. FUNDAMENTOS TEÓRICOS 4](#_Toc76743423)

[El Procesamiento del Lenguaje Natural (PLN) 4](#_Toc76743424)

[Transformers en Procesamiento del Lenguaje Natural 4](#_Toc76743425)

[FINETUNING 5](#_Toc76743426)

[GPT2 5](#_Toc76743427)

[8. DATASET 6](#_Toc76743428)

[9. MÉTRICA DE RENDIMIENTO (FRÉCHET INCEPTION DISTANCE - FID) 6](#_Toc76743429)

[Perplexity 6](#_Toc76743430)

[10. DESCRIPCIÓN DEL TRABAJO REALIZADO EN EL MODELO (MATERIALES Y MÉTODOS) 7](#_Toc76743431)

[Herramientas utilizadas (Software y Hardware) 7](#_Toc76743432)

[Modelos utilizados o arquitectura y descripción del modelo generado (esquemas y/o gráficas). 8](#_Toc76743433)

[HiperParámetros 9](#_Toc76743434)

[BachSize 9](#_Toc76743435)

[Optimizador Adamw 9](#_Toc76743436)

[Learning Rate 10](#_Toc76743437)

[11. SOFTWARE DE APLICACIÓN DESARROLLADO (MÉTODOS Y MATERIALES) 10](#_Toc76743438)

[12. RESULTADOS FINALES 11](#_Toc76743439)

[13. CONCLUSIONES 11](#_Toc76743440)

[14. RECOMENDACIONES 11](#_Toc76743441)

[15. REFERENCIAS BIBLIOGRÁFICAS 11](#_Toc76743442)

# RESUMEN

Este proyecto busca demostrar los alcances de la Inteligencia Artificial aplicándola a un área artística, la poesía. Para ello se usa la red transformas de huggingface que haciendo FineTune es estrenado con datos pre procesados de un conjunto de estrofas y devuelve su composición en datos numéricos para después de un post procesamiento convertir a texto.

# ANTECEDENTES

Los poemas fueron una de las primeras maneras en las cuales los grandes literatos de la historia expresaron sus sentimientos y emociones mediante la palabra escrita. Un poema es considerado tal, en la actualidad, cuando está escrito en verso, a diferencia de aquellos textos escritos en prosa.

Como podría ser por ejemplo un cuento o una novela. El poema tiene a su vez otras características que lo vuelven un género propio y que merece, en quienes los escriben, una habilidad especial.

La importancia de la poesía radica principalmente en que nos permite ampliar nuestros horizontes. Leer poesía puede servirnos para entender mejor el mundo o para buscar nuevas formas de decir las cosas.

Unos de los mayores retos de la Inteligencia Artificial están ligados con la creación de modelos capaces de realizar esas tareas que son propias de los humanos, y en uno de los últimos avances de la misma, se han desarrollado las redes neuronales tipo transformer que son una clase reciente de redes neuronales para secuencias, basadas en la autoatención, que han demostrado estar bien adaptadas al texto y actualmente están impulsando importantes avances en el procesamiento del lenguaje natural.

# PROBLEMÁTICA

Se cree que composición está limitada únicamente a los humanos por su contenido emotivo, pero con composiciones de la IA que logren engañar a los melómanos, ésta se encargaría de la industrialización de la música, mientras los artistas pueden dedicarse más a experimentar y jugar con antiguos y nuevos elementos musicales.

# PROPUESTA

La propuesta es un sistema que generar poemas Mediante el uso de Redes Neuronales tipo Transforme capaz de generar poesía partir del entrenamiento con una gama de Poemas que tengan características similares entre sí, tales como el género, estructura, autor, etc.

# OBJETIVO GENERAL

Utilizar la Inteligencia Artificial para la generación de poesía coherente,

# OBJETIVOS ESPECÍFICOS

* Recopilar un banco de datos
* Hacer una preparación a los datos para su implementación.
* Poder generar coherencia en el texto generado
* Hacer una evaluación del modelo al modelo.
* Implementar el modelo.

# FUNDAMENTOS TEÓRICOS

## El Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una de las ramas del Machine Learning que más atención ha recibido en los últimos años, tanto por parte de la comunidad científica como por parte de las empresas, que ven en sus técnicas un arma para diferenciarse de la competencia, automatizar procesos, mejorar sus productos y conocer mejor a sus clientes, entre otros.

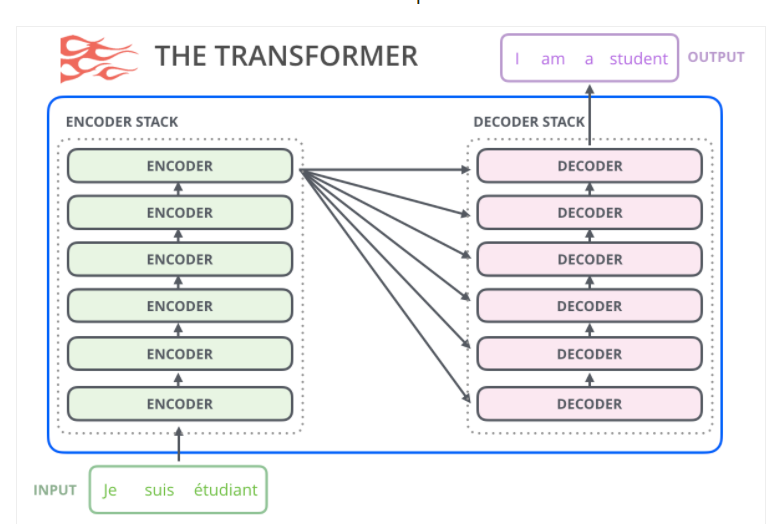
Algunas de las aplicaciones del PLN más conocidas son la generación de textos, detectar entidades (Named Entity Recognition o NER) o clasificar textos, aunque hay muchas más. Esta versatilidad del NLP, y su aplicabilidad a múltiples problemas del mundo real en numerosas disciplinas, resalta la importancia de estas técnicas para diferentes sectores. Además, la investigación en nuevos modelos, especialmente los Transformers, ofrece nuevas posibilidades y está mejorando sus resultados al tener en cuenta el contexto.

## Transformers en Procesamiento del Lenguaje Natural

En un paper de Google de finales del año 2017 -“Attention is All You Need”- se presentó la arquitectura del Transformer, un modelo que tenía como principal innovación la sustitución de las capas recurrentes, como las LSTMs que se venían usando hasta ese momento en PLN, por las denominadas capas de atención.

Estas capas de atención codifican cada palabra de una frase en función del resto de la secuencia, permitiendo así introducir el contexto en la representación matemática del texto, motivo por el cual a los modelos basados en Transformer se les denomina también Embeddings Contextuales.

La arquitectura de Transformer incluye otras innovaciones, como los embeddings posicionales, que permiten al algoritmo conocer la posición relativa de cada palabra del texto. Además, en el paper original se examina la aplicación de esta arquitectura a tareas de traducción, demostrando ser mucho más efectiva que los métodos anteriores en este aspecto, como se puede ver en el gráfico inferior.



## FINETUNING

FINE TUNING significa tomar pesos de una red neuronal entrenada y usarla como inicialización para un nuevo modelo que se entrena con datos del mismo dominio.

 Es usado para:

acelerar el entrenamiento

superar el tamaño pequeño del conjunto de datos

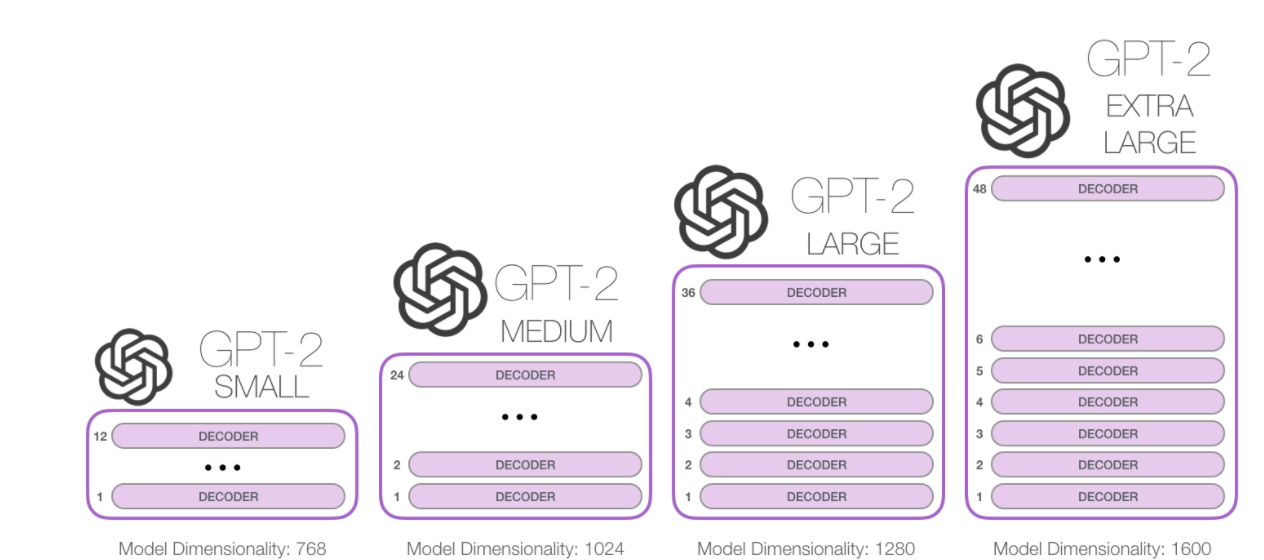
## GPT2

Entonces, ¿qué es exactamente un modelo de lenguaje?

¿Qué es un modelo de lenguaje?

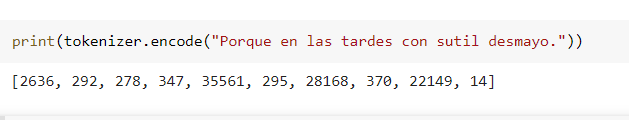
En The Illustrated Word2vec , hemos analizado qué es un modelo de lenguaje, básicamente un modelo de aprendizaje automático que puede mirar parte de una oración y predecir la siguiente palabra. Los modelos de idiomas más famosos son los teclados de teléfonos inteligentes que sugieren la siguiente palabra en función de lo que ha escrito actualmente.

En este sentido, podemos decir que el GPT-2 es básicamente la función de predicción de la siguiente palabra de una aplicación de teclado, pero una que es mucho más grande y sofisticada que la que tiene su teléfono. El GPT-2 se entrenó en un conjunto de datos masivo de 40 GB llamado WebText que los investigadores de OpenAI rastrearon desde Internet como parte del esfuerzo de investigación. Para comparar en términos de tamaño de almacenamiento, la aplicación de teclado que uso, SwiftKey, ocupa 78 MB de espacio. La variante más pequeña del GPT-2 entrenado ocupa 500 MB de almacenamiento para almacenar todos sus parámetros. La variante de GPT-2 más grande tiene 13 veces el tamaño, por lo que podría ocupar más de 6,5 GB de espacio de almacenamiento.



# DATASET

Se tienen 10929 estrofas de poetas recomidos Pablo Neruda, Rubén Darío, Armnado Nervo cuidando las estrofas respetas los 4 versos (Cuarteto) y terminacionen silabas en el formato Soneto ABBA ABBA AABB Con la Tokenizader proporcionado por datificate/gpt2-small-spanish se convierten estos a tokens



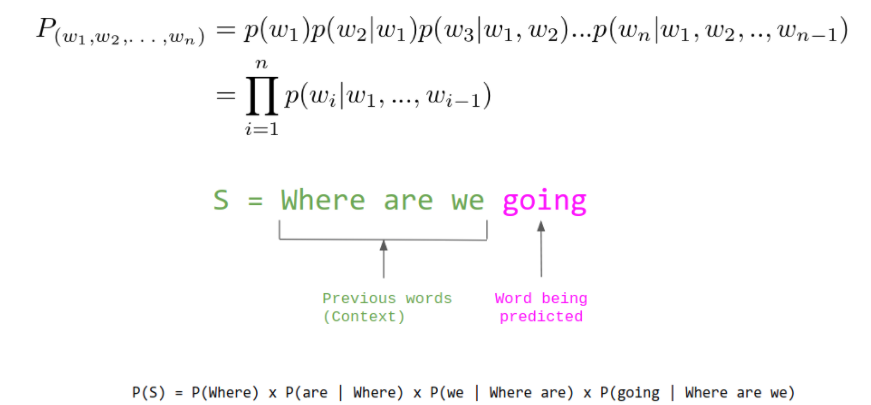
# MÉTRICA DE RENDIMIENTO (FRÉCHET INCEPTION DISTANCE - FID)

## Perplexity

Recientemente, los modelos de lenguaje entrenados en redes neuronales, como ULMFIT, BERT y GPT-2, han tenido un éxito notable cuando se transfieren a otras tareas de procesamiento del lenguaje natural. Como tal, ha habido un interés creciente en los modelos de lenguaje.

Tradicionalmente, el rendimiento del modelo de lenguaje se mide por la perplejidad

Un modelo de lenguaje asigna probabilidades a secuencias de símbolos arbitrarios de modo que cuanto más probable sea una secuencia(w1,w2,...,wnorte)es existir en ese idioma, mayor es la probabilidad. Un símbolo puede ser un carácter, una palabra o una subpalabra (por ejemplo, la palabra "va" se puede dividir en dos subpalabras: "ir" e "ing"). La mayoría de los modelos de lenguaje estiman esta probabilidad como un producto de la probabilidad de cada símbolo dados sus símbolos precedentes:

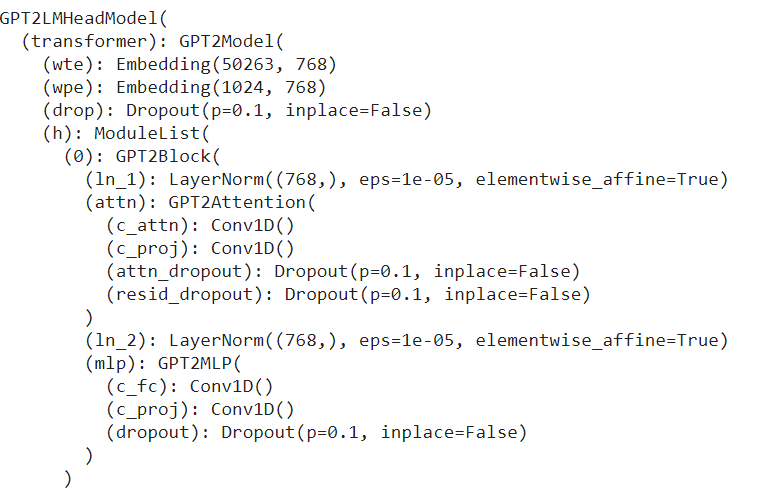


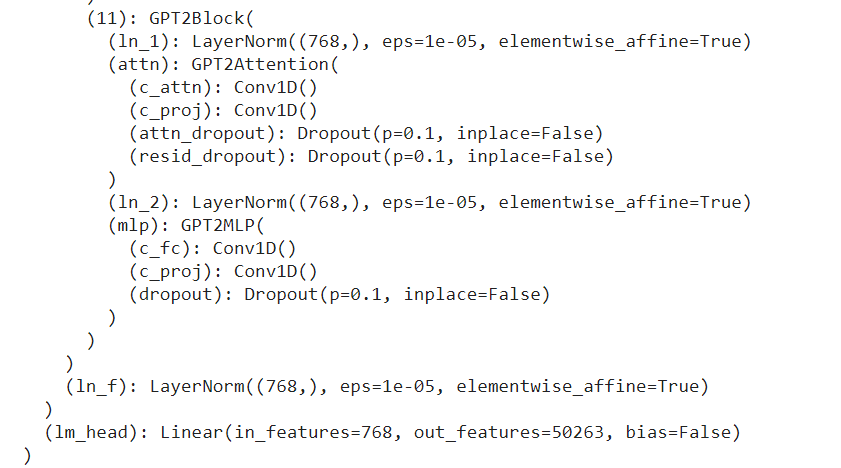
# DESCRIPCIÓN DEL TRABAJO REALIZADO EN EL MODELO (MATERIALES Y MÉTODOS)

## Herramientas utilizadas (Software y Hardware)

* Se utilizará Python (v=3.8.11) como lenguaje de programación.
* Se utilizará como plataforma hasta la fase de entrenamiento y evaluación del modelo, la herramienta Colaboraty Pro de Google.
* Se utilizará la librería Pytorch para la implementación de la red neuronal.
* GTP-2 para la generación de textos y para el Fine Tuning.

## Modelos utilizados o arquitectura y descripción del modelo generado (esquemas y/o gráficas).



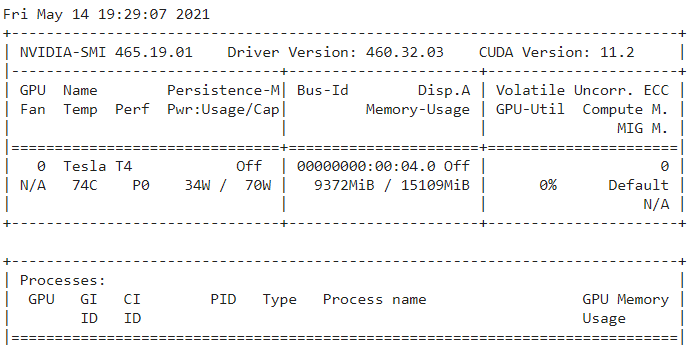


## HiperParámetros

Para el entrenamiento del modelo se utilizaron los siguientes hiperparámetros un learning rate de 0.0001 batchsize de 32, la dimensión de entrada es 100(1024 por defecto), y 4 epochs, siendo éste el valor con el que se evitan sobreajuste al modelo.

### BachSize

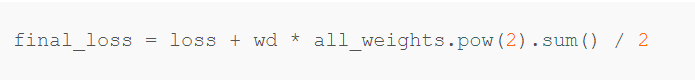
Los diferentes modelos de lenguaje requieren diferentes cantidades de memoria para contener todos los pesos y sesgos en la memoria. Según la memoria que tenga disponible su máquina, esto determinará cómo establecerá el tamaño de su lote.



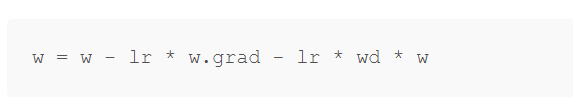
### Optimizador Adamw

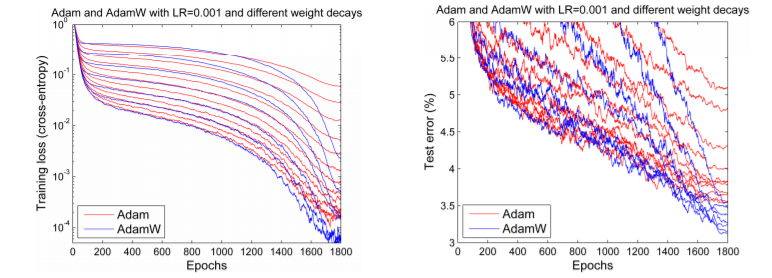
Pero a fines de 2017, Adam parecía tener una nueva vida. Ilya Loshchilov y Frank Hutter señalaron en su artículo que la forma en que se implementa la disminución del peso en Adam en todas las bibliotecas parece ser incorrecta y propusieron una forma simple (a la que llaman AdamW) de solucionarlo. Aunque sus resultados fueron ligeramente mixtos.

La regularización L2 es un método clásico para reducir el sobreajuste, y consiste en agregar a la función de pérdida la suma de los cuadrados de todos los pesos del modelo, multiplicada por un hiperparámetro dado.



Donde w es el hiperparámetro a establecer. Esto también se llama pérdida de peso, porque cuando se aplica vainilla SGD es equivalente a actualizar el peso de esta manera:





### **Learning** Rate

Los pesos de una red neuronal no se pueden calcular mediante un método analítico. En cambio, los pesos deben descubrirse mediante un procedimiento de optimización empírico llamado descenso de gradiente estocástico.

Cada actualización de los pesos realizada en la propagación hacia atrás se pondera con un parámetro llamado tasa de aprendizaje. Si es más pequeño, está avanzando con un tamaño de paso más pequeño hacia el mínimo.

A veces, si está sobreajustado, se recomienda disminuir la tasa de aprendizaje (y aumentar el número de épocas), pero también hay otras formas de combatir el sobreajuste. Una buena tasa de aprendizaje debería permitirle alcanzar un buen mínimo en un tiempo adecuado. Esto es algo con lo que necesitas experimentar.

# SOFTWARE DE APLICACIÓN DESARROLLADO (MÉTODOS Y MATERIALES)

El modelo puede ser usado desde una página web desarrollada con el framework minimalista para Python, Flask.

Este micro framework puede ser aplicado en cualquier patrón de diseño, en este proyecto se aplicará el Modelo Vista Controlador (MVC), por su manejo de rutas.

La exportación e importación del modelo entrenado se realizó con “traicing”, pero los valores que se obtienen son netamente numéricos, necesitan una post procesamiento, para ello se exportó el diccionario generado en el preprocesamiento transformándolo en un archivo JSON para consumirlo desde la página web, permitiendo la conversión a audio.

# RESULTADOS FINALES

El resultado generado son una serie de estrofas de un poema ingresando una entrada la primera secuencia de esta

# CONCLUSIONES

Si es verdad que gpt2 es un modelo robusto su implantación requiere de gran equipo de cómputo, aclarar que para este proyecto se usó el gpt-2-small,

También recalcar los grandes pasos que da la inteligencia artificial siendo los transformes un ejemplo de la evaluación de los diferentes modelos de Deep laringe.

# RECOMENDACIONES

Para poder llevar el proyecto es necesario mucha dedicación al proyecto, leer documentación

# REFERENCIAS BIBLIOGRÁFICAS

* arXiv:1907.00151v5 [cs.CL] 5 Sep 2019 GPT-based Generation for Classical Chinese Poetry
* https://openai.com/blog/gpt-2-1-5b-release/
* https://www.fast.ai/2018/07/02/adam-weight-decay/#understanding-adamw-weight-decay-or-l2-regularization