**Entregable Track A**

**Proyecto:** Implementación de un Traductor en tiempo real de español a quechua para contribuir en las actividades médicas en zonas rurales del Perú

**Equipo / Integrantes:**

* Herrera Cubas Fernando Andrés
* Ramos Maldonado, José Carlos

**Fecha de entrega:** 25-09-2025

1. **Repositorio ordenado y funcional**

**Estructura mínima:**

* 1. **Ejecución del Modelo Bart y Fine tuned**

|  |  |
| --- | --- |
| BART/  │  │  ├── data/  │ ├── somosnlp-hackathon-2022/  │ ├── corpus\_total\_combinado.txt  │ ├── corpus\_total\_combinado.zip  │  ├── modelo\_bart/  │ ├── config.json  │ ├── generation\_config.json  │ ├── merges.txt  │ ├── model.safetensors  │ ├── special\_tokens\_map.json  │ ├── tokenizer\_config.json  │ ├── training\_args.bin  │ └── vocab.json  │  ├── notebooks/  │ ├── 01\_Generacion\_Corpus.ipynb  │ └── 02\_BART\_Qechua\_servidor.ipynb | # Carpeta que contiene datasets y corpus para el modelo  # Subcarpeta con material de un hackathon NLP 2022  # Archivo de texto con el corpus completo combinado  # Archivo comprimido con el corpus completo  # Carpeta del modelo BART entrenado o ajustado  # Configuración general del modelo  # Configuración específica para generación de texto  # Archivo de merges usado por el tokenizador (Byte-Pair Encoding)  # Pesos del modelo en formato safetensors  # Mapa de tokens especiales del tokenizador  # Configuración del tokenizador  # Argumentos de entrenamiento serializados  # Vocabulario del tokenizador  # Carpeta con Jupyter notebooks de experimentos y entrenamiento  # Notebook para la generación de corpus  # Notebook para ejecutar BART en un servidor (para Quechua) |

* 1. **Ejecución de Prototipo (lectura del modelo entrenado)**

|  |  |
| --- | --- |
| PROTIPO\_USUARIO/  │  **├── app/**  │ ├── \_\_pycache\_\_/  │ ├── static/  │ │ └── Logo\_UNI.png  │ ├── templates/  │ │ └── index.html  │ **├── app.py**  **│ ├── ngrok.exe**  **│ ├── run\_with\_ngrok.py**  │ └── **translator.py**  │  ├── **bart\_traducido\_final/**  │ ├── config.json  │ ├── generation\_config.json  │ ├── merges.txt  │ ├── model.safetensors  │ ├── special\_tokens\_map.json  │ ├── tokenizer\_config.json  │ ├── training\_args.bin  │ └── vocab.json  │  ├── env/  │ ├── Lib/  │ ├── Scripts/  │ └── share/  │  ├── .gitignore  ├── CACHEDIR.TAG  └── pyvenv.cfg | **# Carpeta principal de la aplicación Flask**  # Archivos compilados automáticamente por Python  # Archivos estáticos (imágenes, CSS, JS)  # Imagen usada en la aplicación  # Plantillas HTML de Flask  # Página principal con el formulario  **# Archivo principal Flask (servidor web)**  **# Ejecutable de Ngrok para exponer el servidor**  **# Script para ejecutar Flask + Ngrok**  **# Script que carga el modelo y hace la traducción**  **# Carpeta con el modelo fine-tuned de BART**  # Configuración principal del modelo  # Configuración de generación (early\_stopping, beams, etc.)  # Reglas de merges del tokenizer  # Pesos del modelo en formato safetensors  # Definición de tokens especiales  # Configuración del tokenizer  # Argumentos usados durante el fine-tuning  # Vocabulario del tokenizer  # Entorno virtual de Python (dependencias aisladas)  **# Librerías de Python instaladas**  # Scripts ejecutables (ej. python.exe, pip.exe)  # Archivos compartidos del entorno virtual  # Archivo para ignorar archivos en Git  # Marca de directorio cacheado  # Configuración del entorno virtual |

**Archivo requirements.txt**

|  |
| --- |
| **# Dependencias entrenamiento modelo**  python==3.12  accelerate>=0.26.0  torch==2.6.0  CUDA==12.4  transformers==4.53.0  tokenizers==0.21.2  regex==2024.11.6  datasets==3.6.0  dill==0.3.8  xxhash==3.5.0  multiprocess==0.70.16  aiohttp==3.12.13  frozenlist==1.7.0  yarl==1.20.1  **# Dependencias de desarrollo prototipo**  python==3.12.3  flask==3.1.1  transformers==4.39.3  requests==2.32.3 |

Falta librerías de las métricas del modelo.

**Detalle de librerías:**

**Dependencias entrenamiento modelo**

* Accelerate: Herramienta de Hugging Face que facilita el entrenamiento distribuido y optimizado en GPU/TPU/CPU.
* Torch: PyTorch, el framework principal para deep learning, entrenamiento de redes neuronales y uso de GPU.
* CUDA: Toolkit de NVIDIA que permite a PyTorch/TensorFlow usar la GPU para acelerar cálculos.
* Transformers: Librería de Hugging Face con modelos preentrenados (BERT, GPT, T5, etc.) para NLP, visión y audio.
* Tokenizers: Librería de Hugging Face para tokenización ultrarrápida de texto antes de pasarlo a los modelos.
* Regex: Implementación avanzada de expresiones regulares, más potente que el módulo estándar re.
* Datasets: Manejo de datasets grandes (NLP, visión, etc.), con descarga, carga y procesamiento optimizado.
* Dill: Sirve para serializar objetos Python complejos (más allá de lo que puede hacer pickle).
* Xxhash: Algoritmo de hash muy rápido, usado en datasets para verificar integridad y caching.
* Multiprocess: Variante extendida de multiprocessing, que permite ejecutar procesos en paralelo con mejor soporte de serialización.
* Aiohttp: Cliente/servidor HTTP asíncrono, usado para hacer requests concurrentes sin bloquear la ejecución.
* Frozenlist: Estructura de datos usada por aiohttp para manejar listas inmutables que mejoran el rendimiento.
* Yarl: Manejo avanzado de URLs (parsing y construcción) usado también en aiohttp.

**Dependencias de Desarrollo de protototipo**

* + Flask: Microframework web en Python, útil para prototipar APIs o apps web ligeras.
  + Transformers: Igual que antes, librería de modelos preentrenados de Hugging Face (versión diferente para prototipado).
  + Requests: Librería sencilla y popular para hacer HTTP requests (GET, POST, etc.).

**README ejecutable**

|  |
| --- |
| ###########################################  # Traductor Automático de español a quechua  ###########################################  Este proyecto implementa el entrenamiento de un modelo BART de arquitectura de  Transformers con el fine tuning con un Corpus y el prototipo visual de utilización del modelo.  ###########################################  ## 1. Instalación  ###########################################  ### Crear un entorno virtual  ```bash  python -m venv venv  venv\Scripts\activate # En Windows  ```  ### Instalar pytorch y habilitar GPU (CUDA)  ```bash  conda install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126  ```  ### Instalar dependencias  ```bash  # Asegúrate de actualizar pip  pip install --upgrade pip  # Librerías de soporte para entrenamiento  pip install accelerate>=0.26.0  pip install torch==2.6.0  # Librerías de procesamiento de lenguaje  pip install transformers==4.53.0  pip install tokenizers==0.21.2  pip install regex==2024.11.6  pip install datasets==3.6.0  # Utilidades para datasets y multiprocesamiento  pip install dill==0.3.8  pip install xxhash==3.5.0  pip install multiprocess==0.70.16  # Librerías asíncronas  pip install aiohttp==3.12.13  pip install frozenlist==1.7.0  pip install yarl==1.20.1  # Dependencias para prototipo Flask  pip install flask==3.1.1  pip install requests==2.32.3  ```  ###########################################  ### 2. Entrenamiento del modelo  ###########################################  ```bash  python BART/01\_Generacion\_Corpus.py  python BART/02\_BART\_Qechua\_servidor.py  ```  ###########################################  ### 3. Ejecutar el prototipo  ###########################################  # Ejecutar el comando para configurar una cuenta de de ngrok en tu máquina local  usando tu authtoken (clave personal).  ```bash  PROTIPO\_USUARIO\app\ngrok config add-authtoken 2zclPU4R5h99aTdsjjvO8KHW8Lf\_3ec76DGsBzDPs8M3Wu5zg  ```  # Ejecutar la app Flask, esto levantará el servicio de manera local (http://127.0.0.1:5000.)  ```bash  python PROTIPO\_USUARIO\app\app.py  ---  # Ejecutar en otra terminal para generar una URL pública  ```bash  PROTIPO\_USUARIO\app\ngrok http 5000  ---  ###########################################  ## 4. Autores  ###########################################  - Fernado Herrera Cubas  - José Carlos Ramos Maldonado  --- |

1. **Dataset documentado**

**Fuente:** [https://huggingface.co/somosnlp-hackathon-2022/](https://huggingface.co/spaces/somosnlp-hackathon-2022/spanish-to-quechua-translation)

**Fecha de descarga:** 28/10/2022

**Tamaño:** 78.5MB, 128.5K párrafos traducidos.

**Variables principales:**

* **Distribución de Longitud de Oraciones**

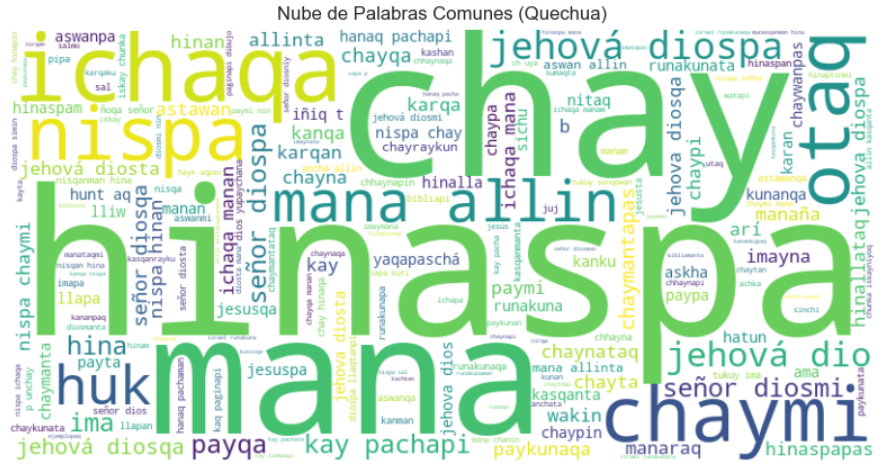
Español: La longitud promedio de las oraciones en español es de 7 palabras. La mayoría de las oraciones se concentran entre 1 y 15 palabras (considerando el percentil 95), con una longitud máxima de 41 palabras.

Quechua: La longitud promedio de las oraciones en quechua es de 6.7 palabras. La mayoría de las oraciones se concentran entre 1 y 14 palabras (considerando el percentil 95), con una longitud máxima de 39 palabras.

* **Valores Nulos / Oraciones Vacías:** Se confirmó que 0 oraciones en español y 0 en quechua resultaron vacías o con 0 palabras después del parsing, lo cual representa un 0.00% para ambos. Esto indica un preprocesamiento efectivo de datos.
* **Palabras Comunes (Ejemplos):**

Español: Las palabras más frecuentes (excluyendo stopwords) incluyen términos como 'salud', 'paciente', 'médico', 'perú', 'hospital'.

Quechua: Las palabras más frecuentes incluyen 'runa', 'llakta', 'qillqay', 'hati', entre otras, que son consistentes con el dominio lingüístico general.

**Hash o control de versiones del Dataset**

Se va utilizar el algorimo Hash para validar el versionamiento del Dataset.

Un algoritmo hash es un procedimiento matemático que toma una entrada (un archivo, un texto, una contraseña, etc.) y produce una salida de longitud fija llamada valor hash o digest.

Para nuestro Dataset al tener peso variable se utilizar el algoritmo SHA-256 (Secure Hash Algorithm 256 bits), genera un digest de 256 bits (32 bytes), que normalmente se representa como una cadena hexadecimal de 64 caracteres.

El algoritmo SHA-256 es **determinístico**, la misma entrada siempre genera el mismo hash; es **no reversible**, no se puede recuperar la entrada original; es **difícil de colisionar**, encontrar dos mensajes con el mismo hash es computacionalmente inviable y tiene **sensibilidad al cambio**, un solo bit cambiado genera un hash completamente distinto (efecto avalancha).

* Preprocesamiento: Se convierte el mensaje (archivo Dataset) en binario, se le agrega un bit 1 y luego ceros hasta que su longitud sea ≡ 448 (mod 512), luego se añade la longitud original del mensaje en 64 bits al final, el mensaje queda dividido en bloques de 512 bits.
* Inicialización: Se definen 8 valores iniciales de 32 bits (constantes derivadas de raíces cuadradas de números primos).
* Compresión: Cada bloque de 512 bits se divide en palabras de 32 bits, se aplican 64 rondas de operaciones (suma modular, rotaciones, desplazamientos, XOR, AND), en cada ronda se usan constantes de 32 bits derivadas de raíces cúbicas de números primos.
* Resultado final: Después de procesar todos los bloques, se obtiene un número binario de 256 bits, para luego expresar en hexadecimal (64 caracteres).

Para nuestro Dataset se utilizará la validación previa al entrenamiento del modelo, en el código Python.

|  |
| --- |
| import hashlib  archivo\_salida = "data/corpus\_total\_combinado.txt"  def generar\_hash\_sha256(ruta\_archivo):  sha256 = hashlib.sha256()  with open(ruta\_archivo, "rb") as f:  for bloque in iter(lambda: f.read(4096), b""):  sha256.update(bloque)  return sha256.hexdigest()  # Ejemplo de uso  archivo = archivo\_salida  hash\_resultado = generar\_hash\_sha256(archivo)  print(f"Hash SHA-256 del archivo '{archivo}':\n{hash\_resultado}") |

**Resultado:**

|  |
| --- |
| Hash SHA-256 del archivo "data/corpus\_total\_combinado.txt":  **3b7e7fed69aeeabb5eb3802c4dd74e6166ddb0c5341f9ee7161068fbc821bc77** |

1. **Pipeline mínimo reproducible**
   1. **Scrip de ingesta y preprocesamiento**
   2. **Logs básicos que documenten el proceso**

**Log ejecución del Modelo Bart y Fine tuned**

**Log ejecución de Prototipo (lectura del modelo entrenado)**

**Tunel para generar URL pública y ejecución de app Flask**

|  |
| --- |
| D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario\app> python .\**app.py**  Traceback (most recent call last):  File "D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario\app\app.py", line 2, in <module>  from translator import translate\_text  File "D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario\app\translator.py", line 8, in <module>  model = BartForConditionalGeneration.from\_pretrained(model\_path)  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  File "C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\modeling\_utils.py", line 3404, in from\_pretrained  model = cls(config, \*model\_args, \*\*model\_kwargs)  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  File "C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\models\bart\modeling\_bart.py", line 1656, in  \_\_init\_\_  super().\_\_init\_\_(config)  File "C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\modeling\_utils.py", line 1237, in \_\_init\_\_  self.generation\_config = GenerationConfig.from\_model\_config(config) if self.can\_generate() else None  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  File "C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py", line 1054, in from\_model\_config  config = cls.from\_dict(config\_dict, return\_unused\_kwargs=False, \_from\_model\_config=True)  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario\app> .\app\**ngrok config add-authtoken**  **2zclPU4R5h99aTdsjjvO8KHW8Lf\_3ec76DGsBzDPs8M3Wu5zg^C**  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario\app> cd ..  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario> .\app\ngrok config add-authtoken 2zclPU4R5h99aTdsjjvO8KHW8Lf\_3ec76DGsBzDPs8M3Wu5zg  Authtoken saved to configuration file: C:\Users\JOSE\AppData\Local/ngrok/ngrok.yml  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario> python .\app\app.py  C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py:526: UserWarning: `num\_beams` is set to None - defaulting to 1.  warnings.warn("`num\_beams` is set to None - defaulting to 1.", UserWarning)  C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py:535: UserWarning: `num\_beams` is set to 1. However, `early\_stopping` is set to `True` -- this flag is only used in beam-based generation modes. You should set `num\_beams>1` or unset `early\_stopping`. This was detected when initializing the generation config instance, which means the corresponding file may hold incorrect parameterization and should be fixed.  warnings.warn(  C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py:535: UserWarning: `num\_beams` is set to 1. However, `early\_stopping` is set to `True` -- this flag is only used in beam-based generation modes. You should set `num\_beams>1` or unset `early\_stopping`.  warnings.warn(  \* Serving Flask app 'app'  \* Debug mode: on  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  **\* Running on http://127.0.0.1:5000**  Press CTRL+C to quit  \* Restarting with stat  C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py:526: UserWarning: `num\_beams` is set to None - defaulting to 1.  warnings.warn("`num\_beams` is set to None - defaulting to 1.", UserWarning)  C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py:535: UserWarning: `num\_beams` is set to 1. However, `early\_stopping` is set to `True` -- this flag is only used in beam-based generation modes. You should set `num\_beams>1` or unset `early\_stopping`. This was detected when initializing the generation config instance, which means the corresponding file may hold incorrect parameterization and should be fixed.  warnings.warn(  C:\Users\JOSE\AppData\Local\Programs\Python\Python312\Lib\site-packages\transformers\generation\configuration\_utils.py:535: UserWarning: `num\_beams` is set to 1. However, `early\_stopping` is set to `True` -- this flag is only used in beam-based generation modes. You should set `num\_beams>1` or unset `early\_stopping`.  warnings.warn(  **\* Debugger is active!**  **\* Debugger PIN: 802-262-287**  127.0.0.1 - - [22/Sep/2025 22:47:12] "GET / HTTP/1.1" 200 -  127.0.0.1 - - [22/Sep/2025 22:47:12] "GET /static/Logo\_UNI.png HTTP/1.1" 200 -  127.0.0.1 - - [22/Sep/2025 22:47:12] "GET /favicon.ico HTTP/1.1" 404 -  127.0.0.1 - - [22/Sep/2025 22:53:30] "GET / HTTP/1.1" 200 -  **127.0.0.1 - - [22/Sep/2025 22:53:30] "GET /static/Logo\_UNI.png HTTP/1.1" 200 -**  **127.0.0.1 - - [22/Sep/2025 22:53:31] "GET /favicon.ico HTTP/1.1" 404 -** |

**Enmascaramiento de la URL privada como pública**

|  |
| --- |
| D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario> **ngrok http 5000**  ngrok : El término 'ngrok' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si  escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e inténtelo de nuevo.  En línea: 1 Carácter: 1  + ngrok http 5000  + ~~~~~  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario> ^C  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario>  \* History restored  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario>  \* History restored  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario>  \* History restored  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario>  \* History restored  PS D:\Jose\UNI\Maestria IA\Cursos\03\_Tercer\_ciclo\03\_Proyecto\_Tesis\Proyecto\Modelo\Protipo\_usuario>  \* History restored  ngrok (Ctrl+C to quit)  Block threats before they reach your services with new WAF actions → https://ngrok.com/r/waf  Session Status online  Account juanjcrm@gmail.com (Plan: Free)  Update update available (version 3.29.0, Ctrl-U to update)  Version 3.23.3  Region South America (sa)  Latency 196ms  Web Interface http://127.0.0.1:4040  Forwarding **https://24c3e486ee70.ngrok-free.app** -> http://localhost:5000  Connections ttl opn rt1 rt5 p50 p90  0 0 0.00 0.00 0.00 0.00 |

