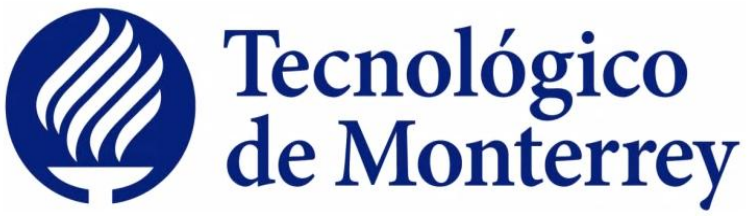

TC5035.10 Proyecto Integrador

Avance 1. Análisis exploratorio de datos

Jesús Ariel Cortés Sarmiento A01552580

Grupo:
45

28 de septiembre del 2025
Monterrey, N.L

Contenido

Transcripción de los archivos de audio

Como primer paso para la creación del sistema RAG, se realizó la transcripción de los archivos .mp4 con Whisper. Este es un modelo Open Source de OpenAI para Reconocimiento Automático de Audio (ASR por sus siglas en ingles). A continuación, se mencionan las principales razones por las que se decidió utilizar este modelo para hacer las transcripciones:

- Es Open Source, lo cual nos ayuda a mantener lo más bajo posible los costos del proyecto, sin tener que sacrificar calidad por ello.
- Cuenta con un fuerte performance para la traducción de audio en español, con cerca de 10k horas de audio en español usados en su entrenamiento (Radford et. al., 2022).

Se realizó la transcripción en Google Colab, ya que esta opción permitía hacer uso de entornos de GPU, para hacer las transcripciones de manera más veloz. De esta manera, se creó una carpeta en Google Drive llamada “RAG Recordings” con las grabaciones de las sesiones de grupo, y luego se accedió a esta carpeta a través de Google Colab.

Se transcribió un solo audio en un principio, para probar con distintas versiones de Whisper cuál daba el mejor trade off de calidad y velocidad. Tras estas pruebas se utilizó la versión “medium” para hacer la transcripción de todos los archivos a través de un loop.

Código

A continuación, se muestra el código que fue utilizado para realizar las transcripciones con Whisper. El archivo .ipynb completo se encuentra adjunto en la tarea como "whisper_transcriptions.ipynb".

```
from google.colab import drive
drive.mount('/content/drive')
```

```
!pip install -U openai-whisper
```

```
100%|████████████████████████████████████| 1.42G/1.42G [00:13<00:00, 114MiB/s]
Collecting openai-whisper
  Downloading openai_whisper-20250625.tar.gz (803 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/803.2 kB ? eta -:--:--
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 798.7/803.2 kB 25.8 MB/s eta 0:00:01
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 803.2/803.2 kB 18.8 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: more-itertools in /usr/local/lib/python3.12/dist-
packages (from openai-whisper) (10.8.0)
Requirement already satisfied: numba in /usr/local/lib/python3.12/dist-packages
(from openai-whisper) (0.60.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages
(from openai-whisper) (2.0.2)
Requirement already satisfied: tiktoken in /usr/local/lib/python3.12/dist-
packages (from openai-whisper) (0.11.0)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages
(from openai-whisper) (2.8.0+cu126)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages
(from openai-whisper) (4.67.1)
Requirement already satisfied: triton>=2 in /usr/local/lib/python3.12/dist-
packages (from openai-whisper) (3.4.0)
Requirement already satisfied: setuptools>=40.8.0 in
/usr/local/lib/python3.12/dist-packages (from triton>=2->openai-whisper) (75.2.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
/usr/local/lib/python3.12/dist-packages (from numba->openai-whisper) (0.43.0)
Requirement already satisfied: regex>=2022.1.18 in
/usr/local/lib/python3.12/dist-packages (from tiktoken->openai-whisper)
(2024.11.6)
```

```
Requirement already satisfied: requests>=2.26.0 in
/usr/local/lib/python3.12/dist-packages (from tiktoken->openai-whisper) (2.32.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-
packages (from torch->openai-whisper) (3.19.1)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.12/dist-packages (from torch->openai-whisper) (4.15.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-
packages (from torch->openai-whisper) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-
packages (from torch->openai-whisper) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages
(from torch->openai-whisper) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages
(from torch->openai-whisper) (2025.3.0)
...
  Stored in directory:
/root/.cache/pip/wheels/61/d2/20/09ec9bef734d126cba375b15898010b6cc28578d8afdde58
69
Successfully built openai-whisper
Installing collected packages: openai-whisper
Successfully installed openai-whisper-20250625
```

```
import whisper
import os
```

```
model = whisper.load_model("medium")
```

```
100%|████████████████████████████████████████| 1.42G/1.42G [00:13<00:00, 114MiB/s]
```

```
def transcribe_audio(file_path: str) -> str:


    result = model.transcribe(file_path, language="es")

    return result["text"]
```

```
file_path = "/content/drive/MyDrive/RAG Recordings/(Audio) Recording 1.m4a"
transcription = transcribe_audio(file_path)
```

```
output_path = file_path.replace(".m4a", ".txt")

with open(output_path, "w", encoding="utf-8") as f:
    f.write(transcription)
```

```
print(f"Transcript saved at: {output_path}")
```

```
audio_folder = "/content/drive/MyDrive/RAG Recordings"
files = os.listdir(audio_folder)
transcriptions = {}
```

```
for file_name in files:
    if file_name.endswith(".m4a"):
        file_path = os.path.join(audio_folder, file_name)

        text = transcribe_audio(file_path)
        transcriptions[file_name] = text
```

```
output_path = os.path.join(audio_folder, "all_transcriptions.txt")
```

```
with open(output_path, "w", encoding="utf-8") as f:
    for file_name, text in transcriptions.items():
        f.write(f"--- {file_name} ---\n{text}\n\n")
```

Referencias:

Radford, A., Wook, J., Xu, T., Brockman, G., McLeavey, C., Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. OpenAI. https://cdn.openai.com/papers/whisper.pdf