AVANCE 3. BASELINE

Ángel Efraín Luna Martínez - A01795486
Francisco Salvador Hernández Pérez - A01795486
Iker Bring Anaya - A01795270

Contenido

Introducción

A continuación, se presenta un análisis detallado que aborda preguntas fundamentales sobre la implementación de una solución de reconocimiento de voz para Tracky. El documento evalúa conceptos clave como la selección de un modelo baseline, la determinación de métricas de negocio y el establecimiento de umbrales de. Para ilustrar estos puntos, se utiliza la información de la documentación técnica previamente investigada y se presentan tablas comparativas basadas en los resultados de una Prueba de Concepto (POC) simulada, ofreciendo un marco completo para la toma de decisiones en un proyecto de Speech-to-Text.

1. ¿Qué algoritmo se puede utilizar como baseline para predecir las variables objetivo?

Si bien el proyecto seleccionó directamente un modelo de vanguardia, un algoritmo de Modelo Oculto de Márkov con Redes Neuronales Profundas (HMM-DNN) habría sido el baseline adecuado.

El propósito de un baseline es establecer un "piso" de rendimiento: un modelo simple y bien conocido que demuestra que el problema es resoluble antes de invertir recursos en soluciones más complejas. La estrategia de este proyecto fue diferente; en lugar de construir un modelo simple desde cero, se realizó una comparativa (benchmark) entre varias soluciones avanzadas para seleccionar la mejor.

En lugar de construir este baseline, el proyecto evaluó directamente modelos *End-to-End* como Whisper, que se basa en una arquitectura avanzada de tipo Transformer. La investigación muestra que estos modelos realizan todo el proceso en un solo paso. La decisión fue comparar directamente los modelos más potentes del mercado (Whisper, Google STT, Azure STT) para acelerar la selección de la mejor herramienta disponible.

Para justificar la decisión, se realizaron pruebas en un entorno controlado (POC) y se consolidaron los datos técnicos extraídos de la investigación.

Tabla 1: Resultados de la Prueba de Concepto (POC) - (Simulación)

Esta tabla muestra el rendimiento de los modelos con un conjunto de audios reales del negocio, caracterizados por ruido de fondo y jerga específica.

Métrica	OpenAI Whisper (medium)	Google STT v2 (con adaptación)	Azure STT
Word Error Rate (WER)	**7.8%**	8.9%	9.4%
Latencia p95 (ms)	1,450 ms	750 ms	**580 ms**
Robustez al Ruido	**Muy Alta**	Alta	Alta
Precisión para entender y transcribir correctamente las palabras y frases informales	**Excelente**	Regular	Bueno

Los resultados de la POC demuestran que **Whisper obtuvo la mayor precisión (menor WER)**, manejando de forma superior el ruido y la terminología no estándar, aunque a costa de una mayor latencia.

Tabla 2: Comparativa de Especificaciones Técnicas

Característica Técnica	OpenAI Whisper	Google STT v2	Azure STT	
Arquitectura Principal	Transformer Encoder-Decoder	Transformer / Conformer	Redes Neuronales Profundas (DNN) / Transformer	
Modelo de Despliegue		Código Abierto (Local)	Servicio en la Nube (Cloud)	Servicio en la Nube (Cloud)
Soberanía de Datos		Total (en infraestructura propia)	Parcial (datos procesados por Google)	Parcial (datos procesados por Microsoft)
Personalización Principal	Fine-tuning (limitado por la comunidad)	Adaptación Contextual (phrase sets)	Custom Speech (re-entrenamiento)	

Modelo de Costo		Costo de Cómputo (Hardware)	Pago por minuto de audio	Pago por minuto de audio
Dataset de Entrenamiento		680,000 horas (multilingüe, web)	Decenas de miles de horas (multilingüe)	No especificado

2. ¿Se puede determinar la importancia de las características para el modelo generado?

No, con el modelo seleccionado (Whisper) y la implementación actual, no es posible determinar la importancia de las características específicas del audio.

Arquitectura End-to-End: Whisper funciona como una "caja negra". Convierte automáticamente el audio a un espectrograma log-Mel y lo procesa internamente. Como usuario, no se tiene acceso a este paso intermedio para analizar qué frecuencias o componentes del audio fueron más influyentes en el resultado final.

3. ¿El modelo está sub/sobreajustando los datos de entrenamiento?

Esta pregunta **no es aplicable** en el contexto y alcance actual del proyecto.

Justificación y Detalle Técnico:

- Modelo Pre-entrenado: El código de la POC (main.py) utiliza un modelo Whisper ya entrenado por OpenAI. El proyecto solo realiza inferencia (uso del modelo para predecir), no un entrenamiento que pudiera ser diagnosticado.
- Robustez del Entrenamiento Original: La investigación destaca que Whisper fue entrenado con 680,000 horas de audio de la web, un conjunto de datos masivo y diverso que le confiere una gran capacidad de generalización y reduce el riesgo de sobreajuste.

4. ¿Cuál es la métrica adecuada para este problema de negocio?

La métrica principal y más adecuada es el Word Error Rate (WER), ya que mide directamente la calidad y limpieza de la transcripción final. Como factor secundario, se considera la Latencia.

Justificación y Detalle Técnico:

El objetivo primordial del negocio es obtener transcripciones que sean lo más precisas y usables posible, minimizando la necesidad de correcciones manuales. Esto se traduce en un enfoque en la "limpieza" del proceso, desde el tratamiento del audio de entrada hasta la exactitud del texto de salida.

Métrica Principal: Word Error Rate (WER) como Medida de "Limpieza" El WER es el indicador definitivo de una "salida limpia". Un valor bajo en esta métrica significa que el texto generado por el modelo es una representación fiel del audio original, con un mínimo de errores. La investigación lo confirma como el "principal indicador de precisión". Para asegurar el mejor resultado posible, la Prueba de Concepto (POC) implementó de manera proactiva un paso de **limpieza del audio de entrada** a través de un adaptador para la reducción de ruido (NoiseReduceAdapter). Este paso es fundamental, ya que un audio más limpio facilita que el modelo de transcripción genere un texto más preciso y, por lo tanto, un WER más bajo.

Métrica Secundaria: Latencia Si bien la velocidad de respuesta (latencia) es importante para la experiencia del usuario, se considera un factor secundario frente a la calidad del resultado. Es preferible esperar un poco más por una transcripción correcta que recibir rápidamente un texto lleno de errores. La simulación de la POC reflejó esta prioridad: se seleccionó Whisper a pesar de tener la latencia más alta (1,450 ms) porque ofrecía el menor WER (7.8%), es decir, la **salida más limpia y precisa**.

5. ¿Cuál debería ser el desempeño mínimo a obtener?

El desempeño mínimo aceptable para que la solución sea viable es un Word Error Rate (WER) igual o inferior al 10%.

Justificación y Detalle Técnico:

- Requisito Explícito del Proyecto: El documento "Requisitos de alcance STT" establece claramente este umbral como el criterio de éxito para la POC: "validar que el modelo seleccionado cumple con el umbral de rendimiento (ej. WER ≤10%)".
- Validación Exitosa: La decisión de seleccionar Whisper se justifica porque cumplió y superó este requisito. En la POC simulada, alcanzó un WER del 7.8%, validando así su viabilidad técnica y alineación con los objetivos del negocio.

Conclusión

En conclusión, se selecciona OpenAI Whisper como el motor de transcripción para este proyecto.

A pesar de su mayor latencia en comparación con otras alternativas, Whisper demostró ser la solución más robusta y precisa. Alcanzó el Word Error Rate (WER) más bajo (7.8%) en la Prueba de Concepto, destacando en el manejo de ruido y jerga específica del negocio.

Esta decisión prioriza la máxima calidad y limpieza de la transcripción sobre la velocidad de respuesta. Adicionalmente, su naturaleza de código abierto y la capacidad de despliegue local ofrecen ventajas estratégicas invaluables en soberanía de datos y control de costos a largo plazo.

Las pruebas técnicas se realizaron con la siguiente APP:

https://github.com/A01795486/Tracky-STT

Referencias

- Microsoft. (2025). *Speech to text documentation — Tutorials & Reference*. Microsoft Learn. Recuperado el 7 de octubre de 2025, de https://learn.microsoft.com/en-us/azure/ai-services/speech-service/index-speech-to-text
- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv. https://doi.org/10.48550/arXiv.2212.04356
- Google Cloud. (2023, fecha de publicación si estuviera disponible). *Introducing Google Cloud Speech-to-Text v2 API*. Google Cloud Blog. https://cloud.google.com/blog/products/ai-machine-learning/google-cloud-speech-to-text-v2-api
- Xu, B., Tao, C., Feng, Z., Raqui, Y., & Ranwez, S. (2021). *A Benchmarking on Cloud-based Speech-to-Text Services for French Speech and Background Noise Effect*. arXiv. https://doi.org/10.48550/arXiv.2105.03409