

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

CÓMPUTO EN LA NUBE



REPORTE DE PROYECTO FINAL

is685434 - Raúl Méndez Álvarez

is708067 - Aristófanés Cruz Huante

Repo:

<https://github.com/RaulMendezA/AWS-Transcribe-Translate.git>

I. Introducción

Las empresas de todo el mundo necesitan formas rápidas y fiables de transcribir un archivo de audio o vídeo y, a menudo, en varios idiomas. Este contenido de audio y video puede variar desde una transmisión de noticias, interacciones telefónicas en un centro de llamadas, una entrevista de trabajo, una demostración de producto o incluso procedimientos judiciales. El proceso tradicional de transcripción es costoso y largo, a menudo implica la contratación de personal o servicios dedicados, con un alto grado de esfuerzo manual, por lo que la idea de poder hacer tal tipo de tareas con servicios de Machine Learning alojados en la nube nos resultó un tanto emocionante desde el inicio.

El procesamiento del lenguaje natural (PNL) y la traducción son algunos de los problemas más difíciles de resolver para las computadoras debido a los muchos elementos contextuales e idiomáticos del habla. Históricamente, esto ha requerido un hablante nativo tanto del idioma de origen como de destino para realizar una traducción. El enfoque basado en redes neuronales recientemente introducido para la traducción automática ha aportado una precisión y fluidez de traducción sin precedentes. Si bien aún no es perfecto, ahora es una solución viable para muchos más casos de uso que nunca.

II. Marco Teórico

Amazon S3 es un servicio de almacenamiento de objetos creado para almacenar y recuperar cualquier volumen de datos desde cualquier ubicación de Internet. Es un servicio de almacenamiento sencillo que ofrece excelente durabilidad, disponibilidad, rendimiento, seguridad y escalabilidad prácticamente ilimitada a costos muy reducidos. [1]

Amazon Transcribe emplea un proceso de aprendizaje profundo conocido como reconocimiento de voz automático (ASR) para convertir audios a textos de manera rápida y precisa. Amazon Transcribe puede usarse para transcribir llamadas del servicio de atención al cliente, automatizar subtítulos y generar metadatos para recursos multimedia y crear un archivo con capacidad completa de búsqueda. Puede utilizar Amazon Transcribe Medical para agregar funcionalidad de texto a audios médicos para aplicaciones de documentación clínica. [2]

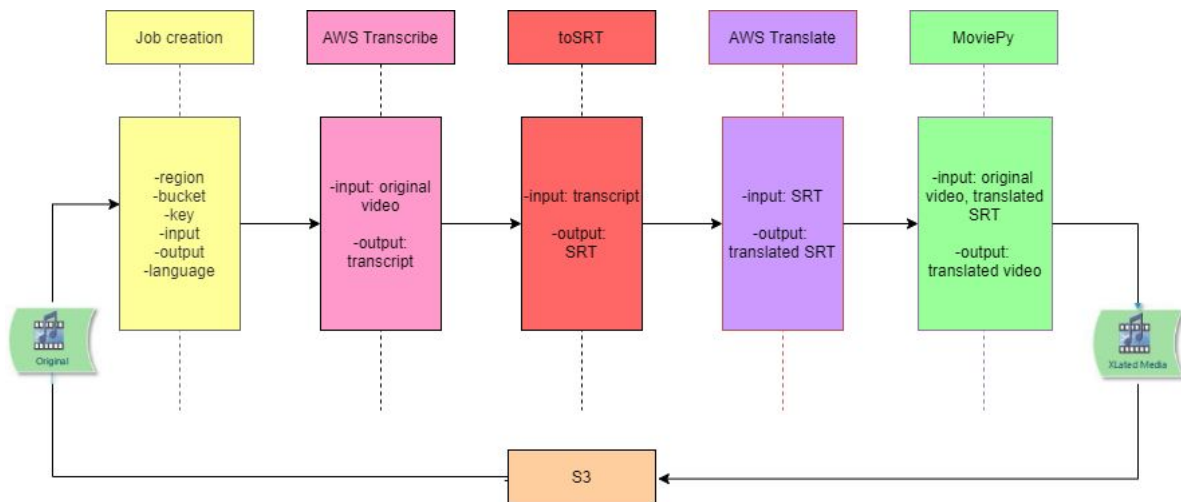
Amazon Translate es un servicio de traducción automática neuronal que ofrece traducción de idiomas accesible, de alta calidad y rápida. La traducción automática neuronal es una forma de automatización de traducciones entre idiomas que usa modelos de aprendizaje profundo para ofrecer traducciones más naturales y precisas que los algoritmos estadísticos tradicionales y de traducción basados en reglas. Con Amazon Translate, puede localizar contenido, como sitios web y aplicaciones para diversos usuarios, traducir grandes volúmenes de texto para análisis con facilidad y permitir una comunicación eficaz entre usuarios de diferentes idiomas. [3]

MoviePy es un módulo de Python para la edición de video, que se puede utilizar para operaciones básicas (como cortes, concatenaciones, inserciones de títulos), composición de video (también conocida como edición no lineal), procesamiento de video o para crear efectos avanzados. Puede leer y escribir los formatos de video más comunes, incluido GIF. [4]

SRT es el formato de archivo empleado por SubRip, un programa que extrae subtítulos desde fuentes de video como video en vivo, archivos de video y discos DVD. SubRip usa tecnología de reconocimiento óptico de caracteres para determinar qué texto de subtítulo es y extraerlo en un archivo de video SRT. El archivo no solo registra el texto en cada subtítulo, sino que también registra los puntos de entrada y salida para que aparezca el subtítulo en pantalla durante la ejecución del video. [5]

III.Desarrollo de la Práctica.

La idea del proyecto fue el poder subir cualquier video a un bucket de S3 y, con la ayuda de diversos servicios (descritos en el párrafo siguiente), detectar la voz y traducir la misma al lenguaje seleccionado para entonces generar un nuevo video con dicha traducción sobrepuesta a manera de subtítulos.



El flujo del proyecto es el siguiente: primero, se genera un comando con la siguiente estructura de argumentos (desglosados como ayuda), y sobre el cual se habrían de obtener los parámetros necesarios para la conversión; se accede entonces al bucket de S3 especificado para leer el archivo de video en cuestión, y con esto es que se crea el job necesario para iniciar el proceso de AWS Transcribe.

```
parser = argparse.ArgumentParser( prog='translatevideo.py', description='Process a video found in
parser.add_argument('-region', required=True, help="The AWS region containing the S3 buckets" )
parser.add_argument('-inbucket', required=True, help='The S3 bucket containing the input file')
parser.add_argument('-infile', required=True, help='The input file to process')
parser.add_argument('-outbucket', required=True, help='The S3 bucket containing the input file')
parser.add_argument('-outfilename', required=True, help='The file name without the extension')
parser.add_argument('-outfiletype', required=True, help='The output file type. E.g. mp4, mov')
parser.add_argument('-outlang', required=True, nargs='+', help='The language codes for the desired
args = parser.parse_args()
```

```
CMD [python translatevideo.py -region us-east-1 -inbucket awscloudproject2020/
-infile AlphabetAerobics.mp4 -outbucket awscloudproject2020/ -outfilename subtitledVideo -outfiletype mp4 -outlang es]
```

CÓMPUTO EN LA NUBE

REPORTE PROYECTO FINAL

Transcribe toma como entrada el video original a convertir, así como el lenguaje de origen ("en-US") y nos genera un Json con el transcript detectado, así como todas y cada una de las palabras presentes, su timestamp y confiabilidad, tal como se ve en la siguiente captura de a continuación.

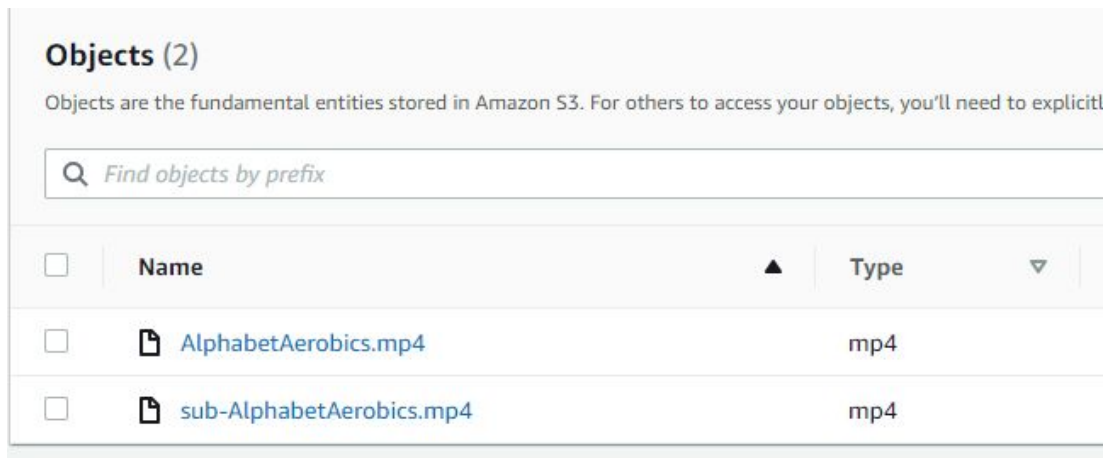
Transcription jobs Info						Download
<input type="text" value="Find job names"/>			Status: All ▾			
	Name	Status	Language	Language settings	Model type	
<input type="radio"/>	transcribe_70e9a20544d545fb9846dcd4feed4ef6_AlphabetAerobics.mp4.mp4	Complete	English, US (en-US)	Specific language	General	



```
{ asrOutput.json > {} results > [ ] items > {} 6
1  {
2    "jobName": "transcribe_89b69610474e4e41b0099a24ba0d3ced_AlphabetAerobics.mp4",
3    "accountId": "980036602778",
4    "results": {
5      "transcripts": [{
6        "transcript": "get artificial amateurs Aren't it all amazing? Analytical
bombarding casually create catastrophes casualties canceling cats got th
other editors with each and every energetic epileptic episode elevated a
goods gone glorious Getting godly in his game with the glorious Hit him
intimidate In an instant I'll rise in an irate state to start my jams li
Khan karate kid type Brits in my kingdom let me live a long life. Lyrica
move many marks. What a master nap No one. I'm nice! Naturally knock nev
perfected poem. Powerful punch lines pummeling petty powder pops in my p
raw wraps rising up rapidly right in the Russian radio activity. Super s
Challengers. Get a tune up. Universal. Unique, untouched. Unadulterated.
with the wise words make just tweeted upwards. We've enough. I'm a worst
mouth young ones Yours yesterday zones ourselves are your six eggs are b
7      }],
8      "items": [{
9        "start_time": "0.04",
10       "end_time": "0.43",
11       "alternatives": [{
12         "confidence": "0.9708",
13         "content": "get"
14       }],
15       "type": "pronunciation"
16     }, {
17       "start_time": "0.44",
18       "end_time": "1.4",
19       "alternatives": [{
20         "confidence": "1.0",
21         "content": "artificial"
22       }],
23       "type": "pronunciation"
24     }, {
25       "start_time": "1.4",
26       "end_time": "2.03",
27       "alternatives": [{
28         "confidence": "0.9885",
29         "content": "amateurs"
30       }],
31       "type": "pronunciation"
32     }, {
33       "start_time": "2.03",
34       "end_time": "2.3",
35       "alternatives": [{
36         "confidence": "0.8092",
37         "content": "Aren't"
```

Esta Json de salida es tomado por Python para transformar el transcript desde una estructura común a un archivo cuya lógica es universal para los reproductores de video en tanto a que éstos pueden tomar el archivo y reproducirlo en sincronía con el video en cuestión. Para esto, primero se forman frases de 10 en 10 palabras, tomando como tiempo inicial al timestamp de la primera palabra, y al final el timestamp de la última.

Este archivo es entonces ingresado a AWS Translate para, frase por frase, traducir su significado al idioma elegido para la conversión. Esto entonces nos genera un segundo archivo de SRT el cual está listo para ser reproducido en conjunto con el video.

La siguiente etapa, y con la ayuda de la librería movie.py, generará el video de salida, para el cual se necesitan el video original y el archivo de los subtítulos traducidos al idioma elegido. La función createVideo entonces renderiza un nuevo video el cual reproduce a ambos en sincronía.



<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	 AlphabetAerobics.mp4	mp4
<input type="checkbox"/>	 sub-AlphabetAerobics.mp4	mp4

La etapa final es simplemente, con las mismas credenciales, transferir o subir el video resultado al bucket de origen, por lo que en la misma plataforma se pueden consultar ambos archivos de inicio y salida. Por otro lado, en el archivo o folder local (incluido en la entrega de este reporte), se generan dichos archivos de subtítulos y video, además de las pistas de audio, una para cada frase transcrita, las cuales habrían de ser empleadas para el servicio de AWS Polly, a modo de voz para ser sobrepuesta en el video de salida - este último paso nos resultó bastante complicado pues, como igualmente se describirá en la sección de problemas, el audio no se podía sincronizar con el video pues el largo en tiempo de cada una de las frases resultaba ser muy distinto de un video a otro.

IV. Problemas y Soluciones

Uno de los principales problemas que tuvimos fue en el translate. Al momento de traducir palabra por palabra y no frases, había mucha diferencia entre el español y el inglés. Ya que en el inglés, lo que podemos decir con una palabra, en el español dice en más. Nuestro programa al momento de hacer las líneas a desplegarse en el video subtitulado, iban de 10 en 10 palabras (en inglés), pero al momento de hacerse la traducción, se extendían esas frases, o en algunos de los casos se acortaban. De ahí surgió el principal problema del uso de Polly, que era la herramienta que queríamos utilizar para que una voz nos tradujera el video y sobreponer el audio. De aquí, las frases traducidas se extendían al momento de hablarse y se atrasaba a lo que iba del video.

El segundo problema con Polly fue que el video al no ser un video de una conversación o audio con flujo “normal” la tecnología Polly se escucha como si se fuera leyendo un texto. No se pudo adaptar a que fuera rapeando como el video lo hacía (aunque hubiera estado genial). Esta tecnología no aceleraba las palabras al ritmo de la canción y solamente se escuchaba como una simple lectura de texto.

Uno de los problemas técnicos con el sistema operativo Windows fue que no se pudo instalar tampoco ImageMagick que trabajaba con Movie.py. Estas dificultades nos acortó a solamente trabajar con el sistema operativo MacOS y hacer las pruebas de la S3 y los trabajos que se estaban procesando desde la otra computadora y no ambas haciendo diferentes pruebas.

V. Experimentos y Resultados.

Como se mencionó en el apartado previo, nuestras primeras pruebas requirieron que empezáramos con probar cada una de las librerías empleadas en el proyecto por separado, probando primeramente con sus features básicos en un inicio y haciéndolo de manera local - esto nos generó muchos problemas al inicio pues, si bien las pruebas resultaron ser sencillas (pues los features de las librerías y servicios de AWS no eran tan complicados como podría sonar), las complicaciones surgieron al tener que hacer las mismas de manera local y en ambientes diferentes (MacOS vs Windows), por lo que optamos mejor por realizar toda la parte local que requirió el proyecto dentro de una misma computadora, para así no tener problemas con las instalaciones de Movie.py e imageMagick.

Comenzamos por videos sencillos de duraciones cortas en español. Estos videos desde el principio en Amazon Transcribe nos arrojaba errores, o mejor dicho, un transcript con palabras con muy baja confianza. Al momento de traducir esto nos daba palabras totalmente fuera de contexto o simplemente dejaba lo que había en el idioma original porque no reconocía la palabra. Optamos por elegir un video donde el inglés se escuchara fuerte y claro, y que el video no tuviera sonidos traslapados. El resultado fue el esperado con el video utilizado para el proyecto.

VI. Conclusiones

Los servicios que ofrece Amazon Web Services nos resultaron bastante interesantes, pues se pueden generar nuevos servicios a partir de combinaciones de los mismos, tal como lo ha sido este proyecto - esto facilita bastante la ejecución de nuevas ideas que se pudiesen tener sobre el desarrollo de propuestas o herramientas en la nube. Aunque es pertinente mencionar que el uso de éstos fue un tanto costoso, pues uno de los miembros del equipo se quedó sin saldo en su cuenta de AWS.

El desarrollo de este proyecto nos permitió experimentar con nuevos servicios dentro de esta misma plataforma, investigar más sobre otros y finalmente decidirnos por un proyecto bueno y funcional que sea útil para algún proyecto futuro. Los servicios de almacenamiento en la nube son muy buena opción para todo este tipo de trabajos que se requiere colaboración de diferentes personas y sistemas.

VII. Bibliografías

- [1] «AWS» [En línea]. Available: <https://aws.amazon.com/es/s3/faqs/>
- [2] «AWS» [En línea]. Available: <https://aws.amazon.com/transcribe/>
- [3] «AWS» [En línea]. Available: <https://aws.amazon.com/es/translate/>
- [4] «Zulko» [En línea]. Available: <https://zulko.github.io/moviepy/>
- [5] Laverty, S. «Techlandia» [En línea]. Available: https://techlandia.com/son-archivos-srt-info_108744/