Universidad Zaragoza
1542

Trabajo Fin de Grado

Transcripción y evaluación automática del freestyle
Automatic transcription and evaluation of freestyle

Autor

Marcos Garralaga Blasco

Director

Carlos Bobed Lisbona

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2023

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./Dª. Marcos Garralaga Blasco ,

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

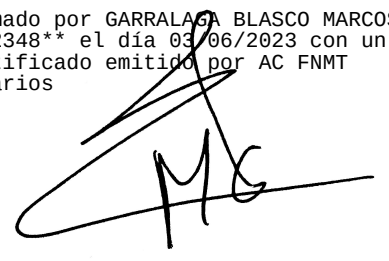
Declaro que el presente Trabajo de Fin de Estudios de la titulación de Grado en Ingeniería Informática (Título del Trabajo)

Transcripción y evaluación automática del freestyle

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 03 de junio de 2023

Fdo: Firmado por GARRALAGA BLASCO MARCOS - ***2348** el día 03/06/2023 con un certificado emitido por AC FNMT Usuarios



AGRADECIMIENTOS

A Carlos Bobed, por su orientación y visión en el proyecto.
A mis padres, por el apoyo incondicional recibido todos estos años.
A Carla, por siempre confiar en mi.

Resumen

Las *batallas de gallos* son una disciplina en la que dos raperos improvisan para comprobar quien es mejor. Si echamos la vista atrás, podríamos asemejar este comportamiento a las disputas entre Góngora y Quevedo, donde resolvían sus diferencias con poesía.

En una *batalla de gallos* hay jueces, los responsables de dictaminar el resultado del encuentro. Este rol puede ser muy complicado, y estar influenciado por las vivencias u opiniones de estos, arrojando en ocasiones juicios subjetivos.

Este trabajo tiene como meta desarrollar una herramienta de evaluación objetiva para la ayuda a estos jueces, aplicando técnicas de transcripción automática de audio a texto, segmentación de este texto obtenido y uso de diferentes conceptos relacionados con el Procesamiento del Lenguaje Natural (PLN) para el análisis de este.

El prototipo desarrollado, compuesto por una API REST y una interfaz web, permite subir un audio y analizar su rima, así como la relación de la letra con conceptos o palabras especificadas por el usuario.

Este trabajo me ha permitido experimentar la realización de un proyecto personal de escala media, con especial énfasis en la organización del tiempo y los esfuerzos, así como la obtención de valiosos conocimientos en el campo de Procesamiento de Lenguaje Natural y procesamiento de audio.

Índice

Capítulo 1

Introducción

Hace unos años conocí el mundo del *freestyle*, una vertiente del rap basada en la improvisación nacida en 1979 en el mundo del *hip hop*, que actualmente cuenta con millones de seguidores. Me convertí en un fanático de esta disciplina, asombrado por la facilidad e ingenio de sus improvisaciones.

Uniendo figuras literarias, estructuras *flow* y *punchlines*, este arte ha sido capaz de ser tratado como deporte, dando lugar a célebres carreras musicales y entreteniendo a sus seguidores. El principal ejemplo de esto son las *batallas de gallos*, una competición en la que dos o más raperos miden quien es el mejor practicando *freestyle*.
Esta competición suele estar dividida en diferentes rondas, en las que los raperos se turnan rapeando con distintos formatos. Estos suelen hacerlo acompañados de un *host*, persona que ayuda al orden y la creación del show; un *DJ*, quien proporciona la música, y un jurado, que puntúa y decide el ganador de la batalla.

Al descubrir las *batallas de gallos*, detecté un general desacuerdo con el veredicto del jurado de estas competiciones ya que este a menudo, puede verse influenciado por sus gustos, vivencias y opiniones.

Pese a que las *batallas de gallos* han experimentado un crecimiento exponencial en popularidad y por ende de formalización en los últimos años, no existe un progreso tecnológico parejo. Los sistemas de puntuación no se han cambiado en más de 10 años, los jueces siguen votando con papel y bolígrafo y existen claros desacuerdos sobre la mejor manera de evaluar el *freestyle*.
Motivado por encontrar una evaluación más objetiva, decidí dedicar mi Trabajo de Fin de Grado (en adelante TFG) a desarrollar una evaluación automática del *freestyle*, fijando el alcance del trabajo con mi director Carlos Bobed para ajustarlo a las horas adecuadas para un TFG.

1.1. Contexto

Este trabajo se trata de un proyecto personal. Desde el primer contacto en mi trayectoria universitaria con el Procesamiento del Lenguaje Natural, me sentí asombrado con la capacidad de la informática para tratar algo tan abstracto como la comunicación. Dentro de los posibles temas a aplicar en mi TFG, este se posicionó entre los favoritos. Armado con la motivación que propicia el deseo de aumentar mis conocimientos, intercambié y debatí varias ideas con mi profesor Carlos Bobed, quien aceptó la dirección de mi TFG, dando lugar a este proyecto.
Decidir el resultado de una *batalla de gallos* es una tarea delicada: los jueces votan siguiendo una normativa de evaluación definida por los responsables del evento, determinando el ganador de grandes premios en metálico. El rol de juez está sometido a una presión constante, en el que el mínimo fallo puede desencadenar consecuencias negativas para este o los responsables del evento[1], y en un ámbito en el que la subjetividad está intrínsecamente ligada a la valoración de los participantes, cualquier aporte objetivo e imparcial es de gran valor para los jueces.

1.2. Objetivos

Con este proyecto se busca evaluar automáticamente un minuto de *freestyle* de la manera más objetiva posible. Esta meta se puede dividir en dos objetivos principales:

1. **Transcripción y obtención de datos**
   El primer paso para lograr la meta fijada es obtener datos de un archivo de audio que representa un minuto de *freestyle*. Para ello, se investigarán diferentes herramientas de transcripción de audio a texto, así como distintas técnicas de separación de texto. La precisión en los datos obtenidos es de gran importancia, ya que sobre estos se basará el análisis y calificación posterior del minuto.

2. **Evaluación del minuto**
   La evaluación es el objetivo final de este proyecto, lo que culminará el trabajo realizado. Se ha decidido seguir una estructura modular, en la que diferentes módulos de evaluación centrados en aspectos específicos del *freestyle* se pueden activar o desactivar, creando así una arquitectura extensible más allá del alcance designado en este TFG.

En la Figura 1.1 se puede observar el proceso que la herramienta seguirá para lograr sus

[1] https://elestilolibre.com/klan-ausencia-supremacia/, accedido por última vez 05/02/23
https://elestilolibre.com/chuty-vs-yenky-bajo-la-lupa/, accedido por última vez 08/02/23

objetivos, comenzando por la transcripción del audio a texto plano, y la segmentación de este en barras[2] y patrones[3], base para la evaluación de la letra.

Figura 1.1: Proceso de transcripción y alineamiento del audio.

1.3. Tecnologías usadas

El sistema operativo utilizado ha sido Linux Mint. Se ha usado el entorno de desarrollo Visual Studio Code, así como el servicio proporcionado por Google Colab, junto con GitHub para el control de versiones. El motor de análisis ha sido desarrollado en Python y se ha empleado el *framework* Angular para crear el prototipo de interfaz. Se han utilizado bibliotecas como Spacy[4], Jiwer[5] y Syltippy[6] para facilitar el Procesamiento del Lenguaje Natural, así como el servicio Web MAUS de CLARIN[7] para realizar el etiquetado temporal de la transcripción .

1.4. Metodología

Una vez presentada y justificada la propuesta así como los objetivos expuestos para el trabajo, a continuación se explicara la metodología que se ha llevado a cabo para la elaboración del TFG. La ejecución del proyecto se ha dividido en intervalos de una o dos semanas en los que se fijaban metas a corto plazo, realizando reuniones con

[2]División estructurada que existe en el rap, comparable a un verso de un poema.
[3]Conjunto de cuatro barras, comparable a una estrofa de un poema.
[4]spacy.io
[5]github.com/jitsi/jiwer
[6]github.com/nur-ag/syltippy
[7]clarin.eu

mi director al final de estas para asegurar el cumplimiento de los objetivos, así como garantizar la correcta dirección del trabajo y obtener consejo sobre este.
Las fases planteadas por las que ha pasado el proyecto son:

1. Creación de un set de datos compuesto por diferentes minutos de *freestyle* para la realización de pruebas.

2. Investigación y análisis de distintas herramientas de transcripción y técnicas de separación del texto en patrones y barras.

3. Implementación de un prototipo de motor de análisis funcional en formato API REST acompañado de un *frontend* web.

4. Desarrollo de diversos módulos de evaluación con sus respectivas pruebas.

En la Sección 5 se puede observar un cronograma detallado de estas fases.

1.5. Organización de la memoria

En este primer capítulo se ha llevado a cabo una pequeña introducción a modo de presentación del trabajo, explicando contexto, objetivos, tecnologías usadas y las diferentes fases del proyecto. En el segundo capítulo se explicará la parte de estudio inicial y análisis del proyecto, la investigación de las tecnologías actuales y las diferentes soluciones. En el tercero, se presentará la implementación del prototipo formado por una interfaz web y una API REST, junto con las decisiones tomadas a lo largo de su desarrollo. En el cuarto, se presentarán los resultados obtenidos, así como posible trabajo futuro, y en el ultimo las conclusiones y la valoración personal del proyecto. Finalmente, en los anexos podremos encontrar un glosario de términos en el que se explican los tecnicismos usados y una pequeña introducción al dominio de las *batallas de gallos*.

Capítulo 2

Investigación y estudio inicial

En este capítulo se presenta el resumen de la fase de investigación realizada para este proyecto. En esta, se analizaron distintas herramientas y se adquirieron los conocimientos necesarios para la realización del trabajo.

2.1. Transcripción

La transcripción de un minuto de *freestyle* es el paso inicial e imprescindible de la investigación. Esta se basa en el paso a texto del audio inicial, que contiene ruido, música y la letra improvisada que se quiere extraer, para aplicar diferentes técnicas de Procesamiento del Lenguaje Natural.

2.1.1. Obtención de datos

Se ha preparado un *set* de datos compuesto por 20 audios de 1 minuto. Estos minutos han sido seleccionados para evaluar la calidad de la transcripción, conteniendo suficiente variación para asegurar la robustez de esta.
Cada minuto ha sido transcrito manualmente con el fin de obtener una evaluación base sobre el que fundamentar nuestra selección.

2.1.2. Herramientas investigadas

A continuación se describen las diferentes herramientas investigadas, así como sus características y observaciones realizadas.

- **AWS Transcribe**: La herramienta de Amazon contiene modelos específicos de aprendizaje automático para transcripciones en diferentes dominios, como un modelo entrenado específicamente para llamadas de teléfono [1] o para consultas médicas [2].
  La transcripción no es la única tarea que puede realizar, también puede identificar automáticamente el lenguaje usado [3] o ejecutar la diarización del hablante [4].
  AWS Transcribe permite una extensa adaptación de las opciones de transcripción, pudiendo establecer un vocabulario personalizado, la eliminación y censura de información delicada, o el entrenamiento y uso de un modelo adaptado al contexto de la transcripción [5].

– **IBM Watson STT**: Esta herramienta creada por IBM también contiene modelos preentrenados, así como la posibilidad de entrenar modelos y ajustar los ya existentes al dominio de uso deseado [6].

  Es capaz de obtener un resultado aproximado muy rápidamente, ya que sus modelos están entrenados para la transcripción de baja latencia [7], en la que se puede observar una transcripción provisional sobre la que la herramienta itera para encontrar el resultado final.

  IBM Watson STT forma parte de un conjunto de herramientas llamado *Watson*, que busca responder a preguntas imitando a los humanos usando PLN [8].

– **Azure STT**: Creada por Microsoft, esta herramienta ataca el problema de la transcripción mediante una combinación de redes neuronales convolucionales (CNN), residuales (ResNet) y *Long short-term memory bidireccional* (Bi-LSTM) [9]. La precisión de esta herramienta es comparable a la de un transcriptor profesional [10], además proporciona una alta robustez frente a ruido en el audio [11].

  Azure también permite adaptar los modelos usados al contexto requerido usando Speech Studio [12].

– **Whisper**: Creada por el equipo de OpenAI, esta se basa en aprendizaje semisupervisado [13] sobre una gran cantidad de datos multilingües y multitarea recopilados de la Web. Whisper es la herramienta más reciente, usando técnicas e innovaciones tecnológicas actuales [14].

  Como curiosidad, comentar que este servicio ha podido transcribir con confianza partes de audio en las que el *ground truth* era dudoso, siendo así la única que ha logrado corregir la transcripción manual en dos ocasiones diferentes.

2.1.3. Estudio comparativo

Se ha realizado un estudio comparativo entre las diferentes herramientas. Para ello se han utilizado diferentes métricas como *Word Error Rate* (WER), *Character Error Rate* (CER), *Match Error Rate* (MER) y *Word Information Lost* (WIL) [15], usando la librería Jiwer:

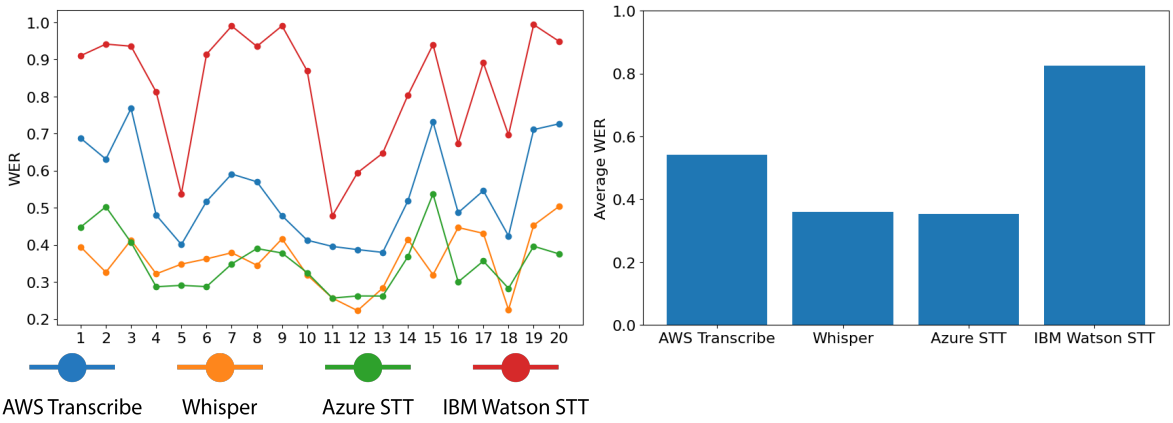
– ***Word Error Rate* (WER)**: El WER se puede expresar como el ratio de errores de una transcripción frente al total de palabras transcritas [16].

$$\text{WER} = \frac{S + I + B}{N} \tag{2.1}$$

  En la Fórmula 2.1 se puede observar como los errores de una transcripción se dividen en 3 tipos:

- Sustitución (S): La herramienta ha usado una palabra errónea, la ha sustituido en la transcripción.
- Inserción (I): La herramienta ha insertado una palabra que no se ha dicho.
- Borrado (B): La herramienta no ha captado una palabra, por lo que la ha borrado de la transcripción.

Esta medida no siempre es equivalente a la precisión de una herramienta de transcripción [17], pero puede otorgar un punto de vista de gran valor.
Se puede observar en la Figura 2.1 un aproximado empate entre Whisper y Azure STT, seguidos de AWS transcribe y por último IBM Watson STT. Un análisis más detallado, usando los percentiles 25 y 75, desvelan que los resultados de Whisper están ligeramente más concentrados, otorgando más consistencia a estos.

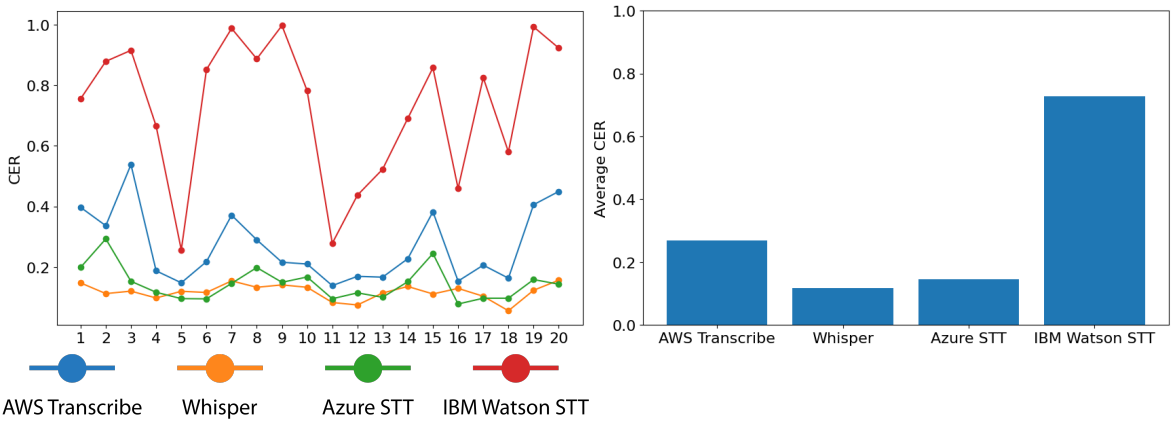


(a) Error en cada audio

(b) Media por herramienta

Figura 2.1: Métricas obtenidas WER.

– ***Character Error Rate* (CER)**: Esta métrica es la misma que WER (Fórmula 2.1), pero opera a nivel de carácter en vez de palabra. Esto permite beneficiar aquellos resultados cuyas palabras erróneas se parecen a las palabras correctas.
En la Figura 2.2 se puede observar otro empate aproximado entre Azure STT y Whisper, pese a que este último ha mejorado notablemente respecto a su WER. Les sigue AWS transcribe con una mejora equiparable y por último IBM Watson STT.
Esta medida se considera de gran importancia, ya que a la hora de evaluar la rima en una palabra fallida se prefieren palabras parecidas a las originales para aproximar los fonemas y así minimizar el impacto de este error.



(a) Error en cada audio (b) Media por herramienta

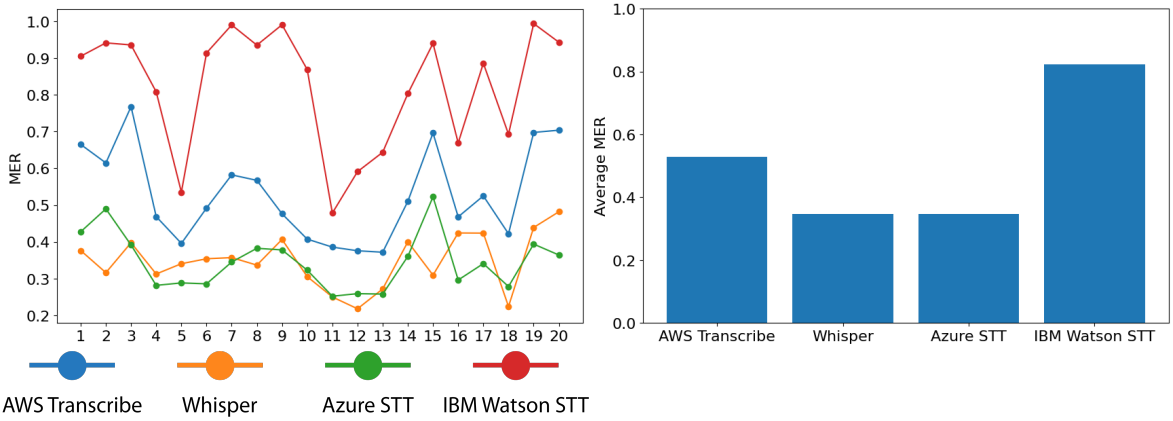
Figura 2.2: Métricas obtenidas CER.

– ***Match Error Rate* (MER)**: El MER se puede resumir como el porcentaje de palabras incorrectamente insertadas, también entendido como la probabilidad de que una palabra sea incorrecta.

$$\text{MER} = \frac{S + I + B}{S + I + B + Hits} = 1 - \frac{Hits}{S + I + B + Hits} \tag{2.2}$$

Debido a su formula matemática, se puede deducir que esta medida siempre será menor o igual que WER (MER $\leq$ WER)

En la Figura 2.3 se puede apreciar como esta medida tiene unos resultados muy parecidos al WER, con un aproximado empate entre Azure STT y Whisper, siendo Azure STT esta vez quien tiene sus resultados ligeramente más concentrados. A estos les siguen AWS Transcribe e IBM Watson STT.



(a) Error en cada audio (b) Media por herramienta

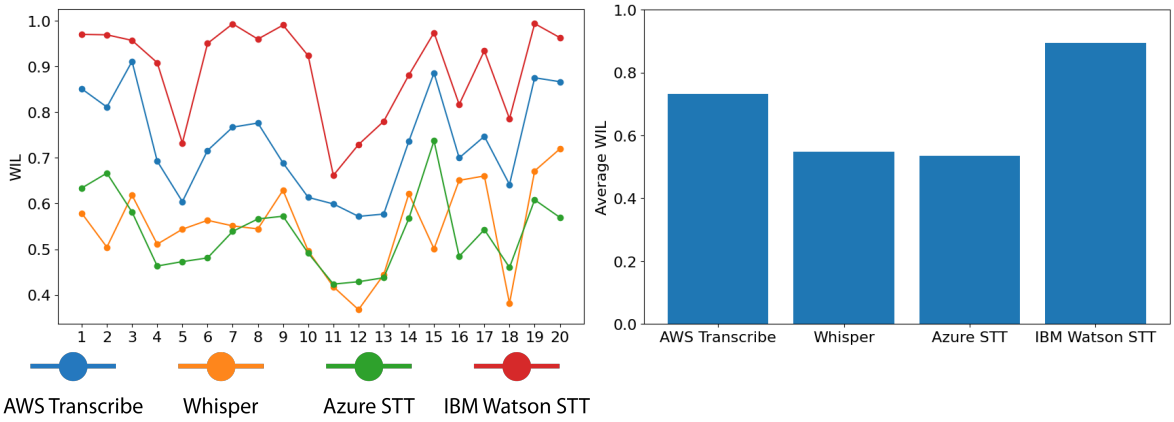
Figura 2.3: Métricas obtenidas MER.

– ***Word Information Lost* (WIL)**: Esta métrica es una aproximación de

Relative Information Lost (RIL) e intenta calcular la dependencia estadística entre la transcripción y el texto real usando solo los *hits*, sustituciones, borrados e inserciones de las palabras [18].

$$\text{WIL} = 1 - \frac{Hits^2}{(Hits + S + B)(Hits + S + I)} \tag{2.3}$$

Pese a que la medida es una aproximación, en esta se puede encontrar el mayor fallo de todas las herramientas, como se demuestra en la Figura 2.4. Azure STT y Whisper vuelven a empatar, seguidos de AWS Transcribe e IBM Watson STT.



(a) Error en cada audio

(b) Media por herramienta

Figura 2.4: Métricas obtenidas WIL.

2.1.4. Resultados

Observamos, según los datos ofrecidos anteriormente, un claro empate entre Whisper y Azure STT. Pese a la gran rapidez de Azure STT, Whisper cuenta con una separación en frases basadas en los silencios y cambios de emisor en la conversación. Esta separación se asemeja bastante a la segmentación en patrones y barras deseada, por lo que se ha elegido Whisper como herramienta de transcripción.
Además, Whisper dispone de diferentes modelos dependiendo de su precisión y rapidez, permitiendo así realizar las pruebas necesarias ágilmente.

2.2. Segmentación en patrones y barras

Como se ha mencionado en la Sección 1.2, un minuto de *freestyle* se divide en patrones y barras, equivalentes a estrofas y versos de un poema. Los raperos deben seguir la música (que no varía en ningún momento) y asegurar que esta división se realiza correctamente, alineando la terminación usada y el tiempo

en el que se construye esta.
En el caso de uso estudiado, un minuto de *freestyle* tiene alrededor de 6 patrones, cada uno compuesto por 4 barras. La segmentación de la transcripción en estas estructuras es importante, ya que permite el análisis posterior de la rima utilizada y de otros aspectos relacionados con la estructura de la letra.
Para obtener esta segmentación se necesita el tiempo exacto en el que se ha dicho cada palabra, por lo que las herramientas investigadas llevan a cabo la tarea de alineamiento de estas palabras transcritas a una línea temporal.

2.2.1. Herramientas investigadas

- **WebMAUS**: Desarrollada por el grupo de investigación CLARIN (Universidad de Munich[1]) recibe como entrada el audio y la transcripción de este, y tras aplicar una serie de tareas de preprocesamiento, traduce la transcripción a una forma fonológica canónica codificada en (X-) SAMPA [19] [20] utilizando la herramienta 'G2P' para usar un modelo de pronunciación probabilística, que encuentra la pronunciación más probable de la señal de voz utilizando la decodificación de Viterbi [21] [22].
  Esta herramienta tiene una granularidad muy fina, pudiendo alinear fonema a fonema. Además, tiene varios formatos de salida, facilitando el procesamiento del resultado.

- **Whisper**: Esta contiene opciones para obtener la alineación realizada. Esta alineación es de palabra a palabra y contiene el punto inicial de esta, así como su duración.

2.2.2. Estudio comparativo

Para obtener la segmentación en patrones y barras se considera que no es necesaria una granularidad muy fina, con obtener el punto medio en el que se ha dicho la palabra es suficiente.
Tras realizar una evaluación manual, se ha observado que WebMAUS se suele equivocar en tramos de alrededor de 3 segundos en los que hay mucho ruido en el audio. Estos tramos suelen ocurrir al final de cada patrón, el momento más importante para la segmentación. Además, se ha observado que el uso de los silencios detectados puede ser de gran utilidad para corregir esta segmentación, así como correcciones del final y del principio de una barra basadas en la rima de estas.

[1] lmu.de/en/

Se ha decidido usar Whisper para la segmentación en patrones y barras debido a su consistencia frente al resto de opciones y la simplicidad otorgada al sistema, haciendo uso de las técnicas comentadas anteriormente. El estudio presentado es menos exhaustivo debido a que la información proporcionada por Whisper se considera suficiente para realizar la alineación del texto, propósito de este.

Capítulo 3

Diseño e implementación de la solución

3.1. Planteamiento del sistema

Para realizar la implementación de la herramienta se ha seguido una arquitectura modular de dos capas basada en servicios, en la que diferentes *plugins* pueden activarse o desactivarse dependiendo de las necesidades del usuario. El sistema esta dividido en dos partes:

- ***Backend***: Compuesto por el motor de cálculo y una aplicación Flask, que expone una interfaz en forma de API REST para la comunicación. Este también contiene cargado en memoria el modelo "*medium*" de Whisper, así como el modelo "*es_core_news_lg*" de Spacy y otros recursos para su uso por los módulos de evaluación.

- ***Frontend***: Interfaz web que se comunica con Backend usando peticiones HTTP, realizada con el *framework* Angular.

En la Figura 3.1 se puede observar la vista lógica de alto nivel del sistema, en la que se distinguen estas dos partes (*Frontend* y *Backend*) unidas por una API REST.

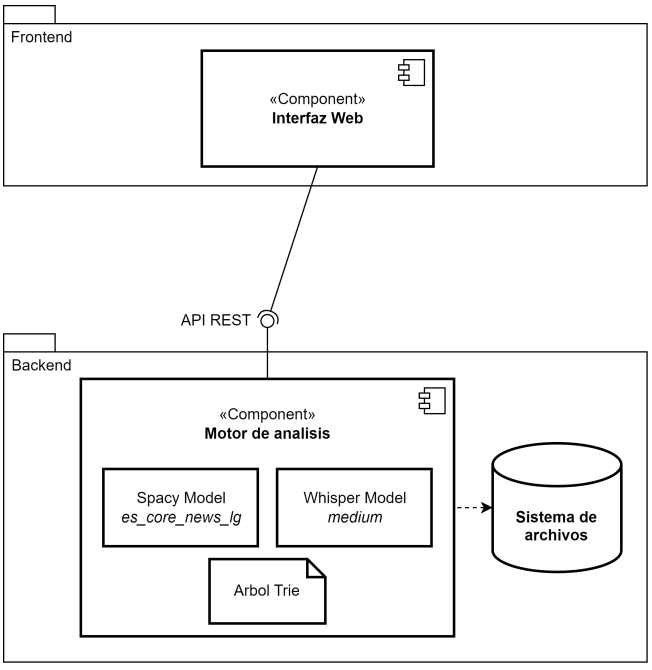


Figura 3.1: Vista lógica.

En la Figura 3.2 se puede apreciar la vista de módulos del sistema, en la que se describen las funciones de los diferentes archivos que forman la herramienta, así como su interacción entre ellos.

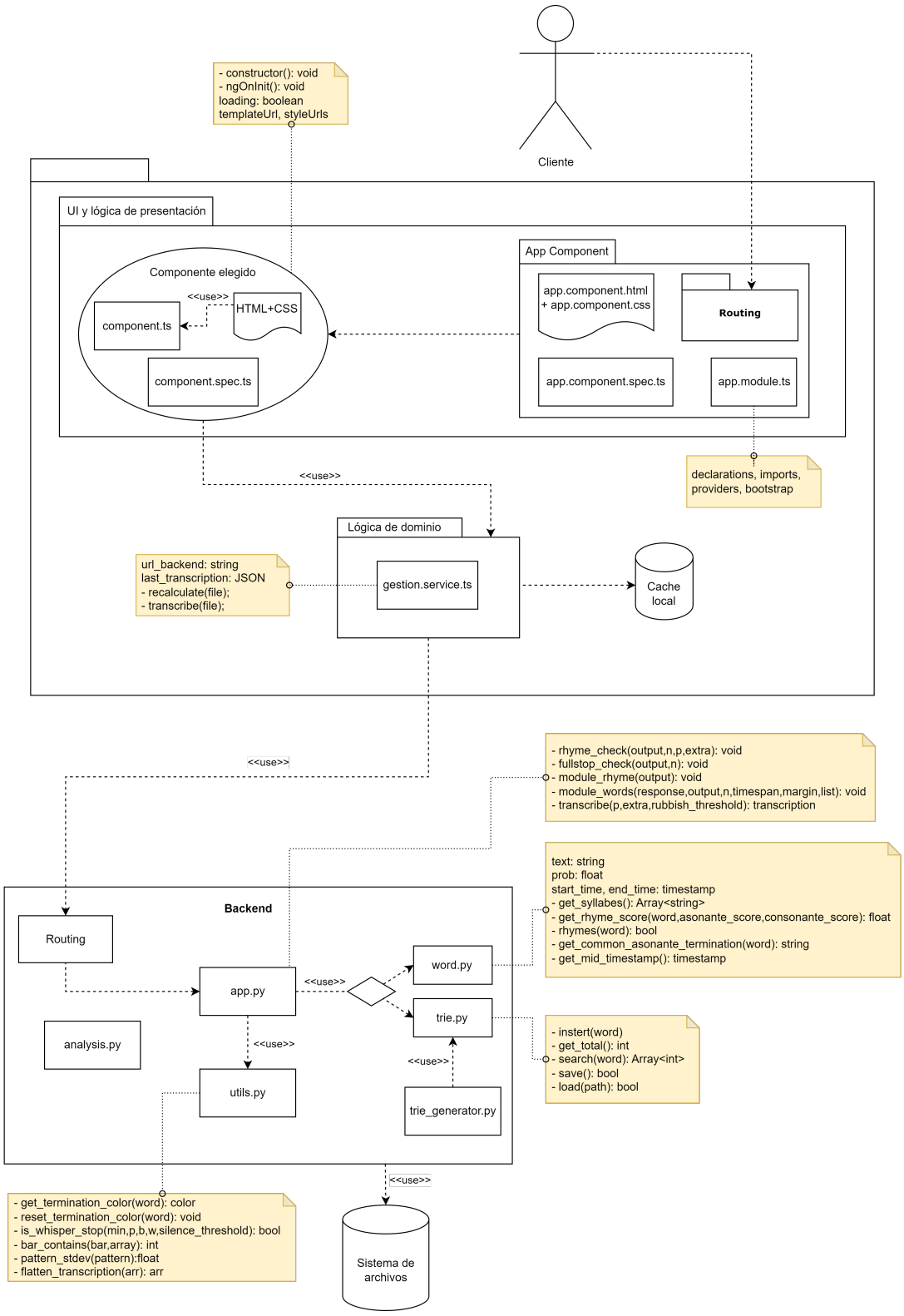


Figura 3.2: Vista de módulos.

3.2. Obtención de datos

A continuación, se describe la implementación de la transcripción y alineación de patrones y barras realizada. Esta se basa en una API REST, accedida por una interfaz web. La implementación puede encontrarse en los archivos adjuntos de este trabajo.

3.2.1. Transcripción

El primer paso para la obtención de datos es la transcripción. Para ello, el motor de análisis carga un modelo de la librería Whisper en Python, que usará para realizar las transcripciones.
Se ha decidido eliminar los acentos de la transcripción. Esto es debido a que el rapero podría jugar con la prosodia como recurso, y la evaluación a realizar no tiene en cuenta estos aspectos, por lo que se ha "normalizado" el texto.
El enunciado se recibe separado en frases, dependiendo del emisor en la conversación y los silencios detectados. Esto será de gran utilidad en el siguiente punto.

3.2.2. Alineamiento de la transcripción

Una parte fundamental de la obtención de datos es la separación en patrones y barras, similares a los versos y estrofas usados en poesía, que constituye la base del análisis realizado.
Esta segmentación se ha realizado siguiendo los siguientes pasos:

- **Eliminación de frases basura**: Es común en la disciplina que el *host*, persona que acompaña a los raperos y ayuda a organizar el show y la batalla, hable o grite mientras se rapea para incentivar al público a apoyar al competidor o transmitir información importante (por ejemplo, informar que es el último patrón). Estas intervenciones tienen varias características únicas: suelen ser cortas (normalmente entre una y tres palabras) y contener interjecciones o palabras clave como "última" o "cambio", además de ocurrir al final de las barras, mientras el competidor toma aire para no solaparse con él.
  Las frases del *host* no aportan ningún valor en la evaluación, por lo que se busca eliminarlas completamente. Para ello, se ha recopilado un conjunto de interjecciones y palabras comunes en estas intervenciones, y debido a la separación en frases de la herramienta utilizada, se puede comprobar si cada frase contiene un ratio alto de estas palabras (parametrizable, por defecto 0.5) para eliminarla de la transcripción.

– **Separación de patrones barras**: Como se ha comentado anteriormente, la música que sigue el competidor para rapear no varía ni se adapta a este, es la tarea del rapero contener su patrón en un intervalo de tiempo. Por consiguiente, podemos encontrar el punto inicial del minuto (primera palabra transcrita) y su punto final (última palabra transcrita). Usando estos dos puntos y conociendo el número de patrones del minuto, se puede realizar una segmentación de las palabras por tiempo en patrones y barras, obteniendo una primera aproximación. La realidad de las *batallas de gallos* es que pocas veces el rapero consigue contener todas sus palabras en el intervalo de tiempo designado (muchas veces solo por décimas de segundo), debido a imprecisiones de este y de los puntos de inicio y final mencionados anteriormente. Por lo tanto, se aplican una serie de correcciones para maximizar la correcta segmentación de la transcripción:

  - **Corrección por silencios**: Esta primera corrección se basa en los silencios detectados por Whisper. Es común que el rapero tome aire entre patrones o barras, tal y como lo realizamos en una conversación entre las frases, por lo que si un silencio se encuentra cerca de una de las fronteras designadas, se puede suponer que la frontera es incorrecta, moviéndola al sitio adecuado. Esta comprobación se realiza en un número de palabras parametrizable (por defecto 3) de cada barra, encontradas al inicio y al final de esta.
  - **Corrección por rima**: Los patrones contienen cuatro barras cada uno, por lo que se puede acotar la estructura usada a los siguientes tipos:
    - Rima continua (AAAA): Todas las barras contienen la misma terminación.
    - Rima gemela o pareada (AABB): Se rima de dos en dos barras seguidas.
    - Rima abrazada (ABBA): Rima la primera y última barra, así como las dos interiores.
    - Rima cruzada (ABAB): Riman los versos impares entre sí, y los pares entre sí.

    Al no conocer la estructura usada, se comprueban todas las estructuras realizando todos los movimientos posibles de las palabras entre barras (fuerza bruta), acotados por el número de palabras máximas a mover de cada barra (parametrizable, por defecto 3). En cada posible configuración de movimiento de palabras, se debe encontrar la estructura utilizada, para lo que se calcula una puntuación de cada una de estas que indica como de probable es que este patrón contenga esa estructura, usando las siguientes medidas:

- Terminación: Tiene el mayor valor, ya que la estructura se basa en la rima usada por el competidor. Para ello, se valoran positivamente aquellas sílabas que tienen una rima consonante y algo menos a aquellas con rima asonante (ambos valores parametrizables).
- Variación de longitud: Suponiendo que las barras tienen un número parecido de sílabas (porque hay un tiempo límite para decirlas y los raperos suelen llevar el mismo ritmo a lo largo de un patrón), cualquier configuración que tenga una longitud de barras desbalanceada debe ser penalizada. Para ello se usa la desviación estándar de la longitud de las sílabas de estas.

Una vez encontrada la configuración del patrón con mayor puntuación, se puede aplicar esta corrección basada en la rima. Cabe destacar que normalmente esta corrección no efectúa ningún cambio, como máximo mueve una o dos palabras. Además, se puede guardar la estructura detectada de cada patrón para su futuro análisis.

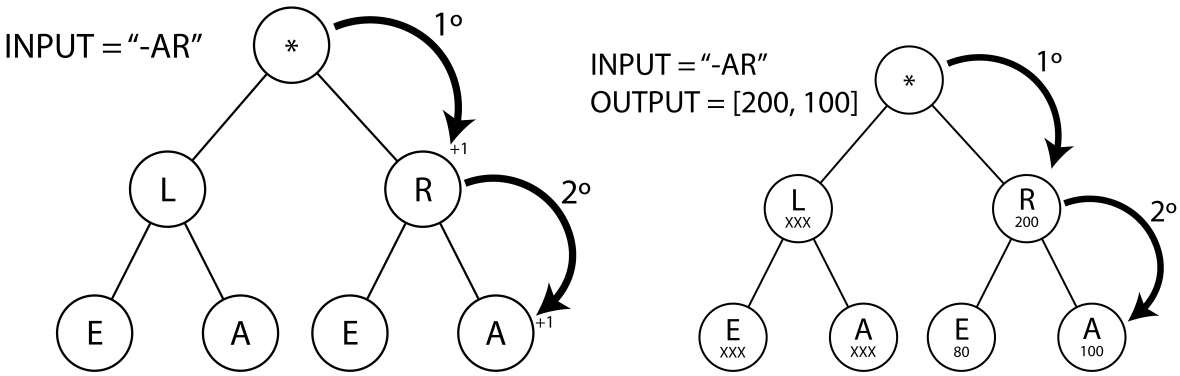
3.3. Módulos de evaluación

La evaluación del minuto de *freestyle* se ha realizado modularmente, dividiéndose en diferentes "módulos de evaluacion", que se ocupan de puntuar una característica específica del *freestyle* (por ejemplo, la terminación usada). Además, estos módulos pueden ser activados o desactivados, dependiendo de las necesidades del usuario.
A continuación se explicarán los módulos desarrollados.

3.3.1. Terminaciones

Este fue el primer módulo desarrollado, se basa en la idea de que las rimas tienen distintas dificultades: aquellas terminaciones que dispongan de pocas palabras españolas son más difíciles de rimar, y por ende mejor valoradas. Para evaluar la rareza de una terminación se necesita un *set* de palabras españolas lo más completo posible. A tal fin, se ha usado el conjunto proporcionado por BabelNet 5.2, de la Universidad Sapienza en Roma. Se ha construido un árbol Trie de las terminaciones invertidas letra a letra, guardando su frecuencia, pudiendo recorrer este árbol para obtenerla. Así se puede calcular una puntuación aproximada de la dificultad de la terminación.

Este módulo evalúa la rima consonante y asonante de cada una de las barras para obtener una puntuación del patrón.



(a) Inserción terminación -ar (b) Puntuación terminación -ar

Figura 3.3: Comportamiento árbol Trie.

3.3.2. Modo palabras

Uno de los recursos más utilizados para comprobar si de verdad se está improvisando son las "palabras": Se trata de la propuesta de uso de una palabra en la intervención del rapero. Este recurso se puede plantear de muchas maneras: una palabra para todo el minuto ("temática"), palabras cada X segundos ("*easy mode*" y "*hard mode*"), etc... Además, está bien visto que el rapero no solo use esta palabra en su improvisación, sino que también utilice su campo semántico, demostrando todavía más que su improvisación es real.

Para evaluar esta relación del patrón a una palabra se han utilizado los *word embeddings*. Un *word embedding* es una representación vectorial densa de una palabra. Aquellas palabras cuyos vectores sean más cercanos tendrán un significado más similar. La librería Spacy contiene modelos de Procesamiento de Lenguaje Natural que permiten el cálculo de estos *word embeddings*.

Normalmente, la distancia entre dos vectores es calculada con la similitud coseno.

$$\text{sim}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|} \tag{3.1}$$

Con esta fórmula se obtiene un valor entre -1 y 1, pero no se busca dar una puntuación negativa a aquellas palabras que no se parecen mucho a la "palabra objetivo", por ello se ha decidido aplicar la distancia angular entre estos dos vectores como alternativa para calcular la similitud semántica de dos palabras [23]:

$$rel_w(A, B) = ang.distance(A, B) = 1 - \frac{arccos(sim(A, B))}{\pi} \tag{3.2}$$

Así se obtiene un valor entre 0 y 1, y tras eliminar los *stop words*, calcular la distancia angular entre los *word embeddings* de las palabras restantes y la suma de los valores mas altos calculados, se puede obtener una puntuación del patrón. Este módulo también permite la parametrización de un bonus por usar la "palabra objetivo".

3.4. Interfaz

Para facilitar el acceso a la herramienta se ha desarrollado un prototipo de interfaz web. En la Figura 3.4 se puede apreciar como esta interfaz permite seleccionar entre dos modos de análisis:

Figura 3.4: Página principal del interfaz

– **Modo normal**: Este recibe un audio, lo transcribe y ejecuta el análisis de la terminación. En la Figura 3.5 se muestra la pagina de la herramienta, en la que se puede modificar el número de barras del minuto a analizar, así como otras medidas.

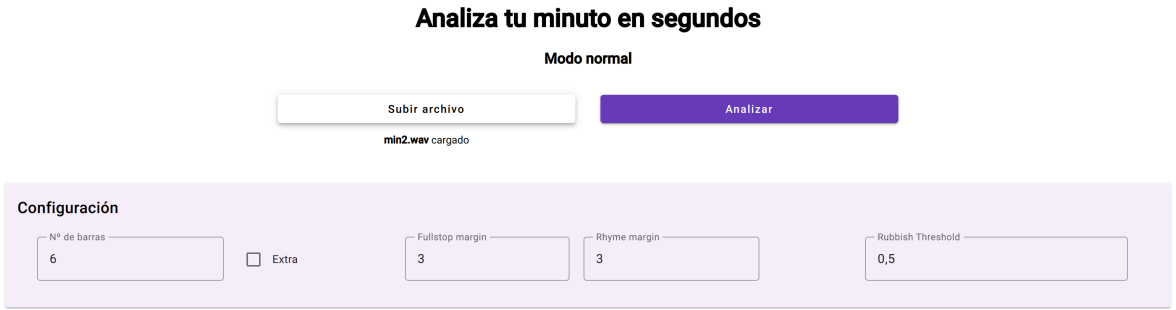


Figura 3.5: Página modo normal

– **Modo palabras**: Este recibe un audio y un conjunto de palabras y transcribe el audio, ejecuta el análisis de la terminación y calcula la puntuación de cada palabra proporcionada con el módulo comentado anteriormente. En la Figura 3.6 se muestra la configuración del modo, en la que se puede modificar el número de barras del minuto a analizar, acompañadas opcionalmente por la duración en el tiempo de cada palabra así como otras medidas.

La salida está dividida en dos zonas:

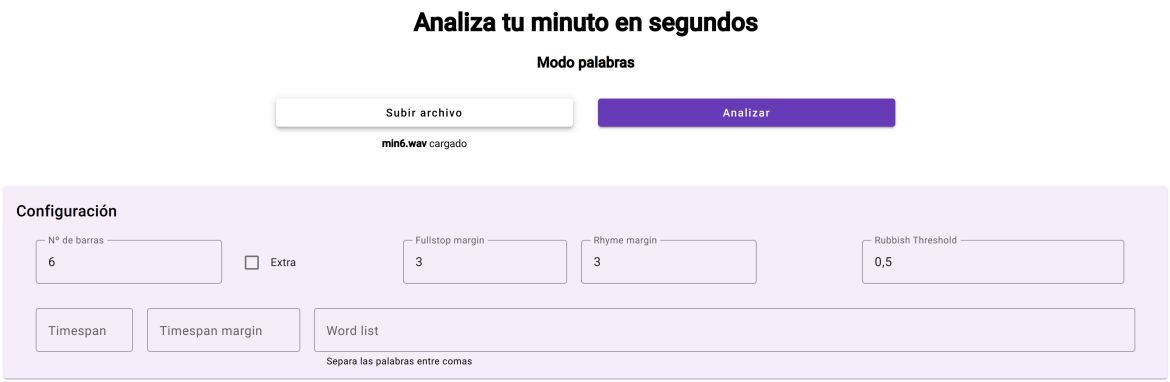


Figura 3.6: Página modo palabras

– La primera, representada en la Figura 3.7a), es común en los dos modos y contiene la transcripción segmentada en patrones y barras, cuya transparencia de las palabras muestra la confianza del modelo de Whisper sobre esa transcripción. Posicionando el cursor sobre una palabra y sin moverlo durante unos segundos permite visualizar el grado de confianza de la misma.

– La segunda, representada en la Figura 3.7b), contiene el análisis de la terminación usada, así como el subrayado de la rima asonante. En el caso del modo palabras, como se puede apreciar en la Figura 3.7c), también contiene la puntuación de cada patrón respecto a la similitud de este a la "palabra objetivo", señalado en color rojo si no se ha utilizado esta.

Transcripción obtenida

Porque es que vengo, hermano, y yo tengo más lésico.

Estratosfera, tengo un nivel estratosférico.

Porque te digo que no me paras mis balas.

Tú no eres mala madre, pero tu madre es mala.

A veces aprendes a hacer los putos patán.

Sabes que ahora mismo tú vas a bailar cancan.

Pero es que muero un payaso.

Si me pongo de puntilla, puedo darte un cabezazo.

Ya sé que eres campeón mundial.

No hace falta que me lo digan

más. Pero este año haré dos cosas que tú no harás.

Una es quedarme en la Liga, otra irá a la Nacional.

(a) Transcripción del audio

Alineación obtenida

Porque es que vengo hermano y yo tengo más lésico	0.16
Estratosfera tengo un nivel estratosférico	0.16
Porque te digo que no me paras mis balas	0.03
Tú no eres mala madre pero tu madre es mala	0.12

Rhyme score: **0.47**

A veces aprendes a hacer los putos patán	0.02
Sabes que ahora mismo tú vas a bailar cancan	0.02
Pero es que muero un payaso	0.15
Si me pongo de puntilla puedo darte un cabezazo	0.15

Rhyme score: **0.34**

(b) Puntuación de terminaciones

Alineación obtenida

Porque es que vengo hermano y yo tengo más lésico

Estratosfera tengo un nivel estratosférico

Porque te digo que no me paras mis balas

Tú no eres mala madre pero tu madre es mala

Rhyme score: **0.47**
Word score: **16.86**

A veces aprendes a hacer los putos patán

Sabes que ahora mismo tú vas a bailar cancan

Pero es que muero un payaso

Si me pongo de puntilla puedo darte un cabezazo

Rhyme score: **0.34**
Word score: **5.12**

(c) Puntuación de palabras

Figura 3.7: Salida de la herramienta.

Capítulo 4

Resultados

En este capítulo se muestran los resultados obtenidos tras el desarrollo de este trabajo. Adicionalmente, se muestran aciertos y fallos observados en el uso de la herramienta. Por último, se comenta el trabajo a futuro del proyecto.

4.1. Ejemplo de uso

La herramienta permite la selección del tipo de análisis a realizar. Como se puede observar en la Figura 3.4, si se trata de un análisis con palabras se deberá seleccionar el modo palabras. De lo contrario, se seleccionará el modo normal.
A continuación se debe subir el archivo de audio que representa el minuto, así como especificar la configuración del motor de análisis, compuesto por el número de barras y, en el caso del modo palabras, las palabras usadas, así como su duración.
Al presionar el botón "Analizar", la interfaz envía una petición a la API REST, que analiza este audio y devuelve los resultados pertinentes.

4.2. Fallos observados

A continuación se detallan los diferentes fallos observados, así como su explicación.

- **Transcripciones incorrectas**: El modelo de Whisper tiene problemas identificando algunas palabras. Este error no se considera de mucho efecto, ya que Whisper proporciona en su intento de transcripción la palabra con mayor confianza, que suele asemejarse morfológicamente a la palabra real, por lo que no se detecta un impacto significativo.
  Algunos ejemplos de estas transcripciones incorrectas se detallan en la Tabla 4.1

  Se pueden observar algunas características de las palabras fallidas: a veces estas se tratan de nombres propios, motes o tecnicismos específicos de la disciplina, por lo que es entendible que Whisper no las conozca. Otra característica común encontrada es el ruido, especialmente presente en aquellos errores en los que las sílabas son muy parecidas, pero la separación en palabras es incorrecta. También se observan fallos en algunos anglicismos, ya que el sistema no está preparado para los cambios de idioma tan repentinos. Además, Whisper muestra problemas

Hipótesis	Referencia
lésico	léxico
aprendes a hacer los	aprendes a hacerlo
que se escrito notas si él ha contestado	que es escrito en notas si era contestado
viento mi tiempo y que aprendan	invierto mi tiempo en que aprendan
a la estúa	al Stuart
Porque se desobra	Porque sé de sobra
que nos critican	que ellos critican
años te pemelece	años de FMS

Tabla 4.1: Fallos observados en la transcripción

en el caso de las composiciones y derivaciones de las palabras, así como en el uso de diptongos y sinéresis. Por último, el sistema encuentra problemas con raperos que no pronuncian mucho sus letras o rapean excesivamente rápido.

- **Alineaciones incorrectas**: El sistema de alineación de la transcripción en patrones y barras se considera muy primitivo comparado con la cantidad de recursos utilizados en el *freestyle* hoy en día: algunos raperos usan "barras vacías" (aquellas en las que se usa el lenguaje no verbal acompañado de silencio) para construir estructuras exóticas compuestas por solo tres barras. Estas se consideran fuera del alcance de este estudio.
  En algunos casos, el sistema segmenta incorrectamente las barras, usando palabras colindantes a la correcta que también riman, por lo que estos fallos se consideran de baja severidad.

4.3. Trabajo futuro

En esta sección se comentará el posible trabajo futuro relacionado con el proyecto, basado en un análisis cualitativo de las capacidades de la herramienta.
Una de las principales tareas a futuro no realizada por falta de tiempo es la evaluación de los resultados obtenidos. Actualmente, solo han sido analizados por mí, considerado experto en el dominio, pero me gustaría contactar con jueces profesionales para asegurar la utilidad de la herramienta y recoger el *feedback* necesario para su mejora, incluyendo la visualización de información que se considere útil para estos, así como el cálculo de nuevas métricas recomendadas.
Al desarrollar el análisis de una manera modular siempre se puede expandir, añadiendo módulos que, por ejemplo, traten la rima interna (rima dentro de cada barra), analicen si un patrón es una respuesta a otro patrón del rival, detecten rimas recicladas (que ya se han dicho en un pasado, por ende penalizadas), avisen a los jueces de la posible

presencia de rimas escritas (el competidor no está improvisando) o evalúen la variación en el ritmo del rapeo, usando la alineación temporal a nivel de fonema proporcionada por WebMAUS.
También me gustaría mejorar la segmentación de la transcripción en patrones y barras, aplicando técnicas de aprendizaje automático que tengan en cuenta el etiquetado gramatical de cada palabra y su posición en el minuto.
A partir de esta propuesta, se podría utilizar la herramienta para tratar otros temas relacionados con la rima, como es la poesía o la letra de una composición musical.

Capítulo 5

Conclusiones

En este capítulo se comentarán las conclusiones del trabajo, basadas en el tiempo empleado y la valoración personal de este.

5.1. Tiempo estimado empleado

En la Tabla 5.1 se presenta el tiempo total empleado, así como su división en diferentes tareas.

Tarea	Horas dedicadas
Análisis del dominio	15
Diseño y formalización de idea	18
Reuniones	16
Investigación inicial	34.5
Creación set de datos	14
Desarrollo transcripción y alineamiento	62
Desarrollo evaluación	59.5
Desarrollo interfaz	10
Tests herramienta	20
Memoria	58
Preparación presentación	15
Total	322

Tabla 5.1: Distribución de horas en tareas

La distribución de estas tareas se puede apreciar en la Figura 5.1

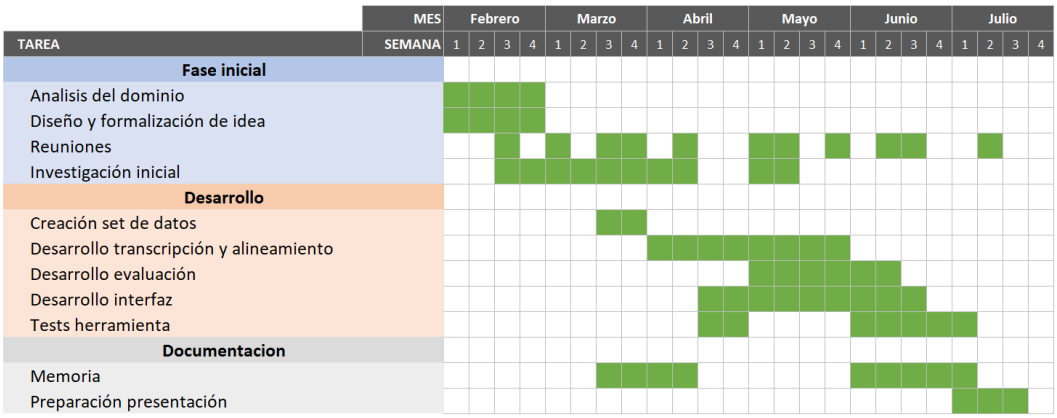


Figura 5.1: Diagrama de Gantt.

5.2. Valoración personal

La elaboración de este trabajo es uno de los primeros contactos con el mundo profesional del PLN, con su correspondiente investigación previa (paso imprescindible en un proyecto de esta índole), así como la aplicación de técnicas recientes en análisis de la transcripción y otros conocimientos adquiridos en el grado.
El hecho de que este TFG se trate de un ensayo personal me ha mostrado lo difícil que es organizar un proyecto desde cero, desde la formación de la idea hasta el desarrollo de un primer prototipo. Compaginar este trabajo con mi beca de colaboración en LIFTEC, me ha permitido desarrollar habilidades organizativas muy valiosas, llegando a este punto con la satisfacción que otorga la finalización de un proyecto de esta magnitud, pese a la dificultad y dureza que acarrea.
Por último agradecer a todo el profesorado su dedicación hacia mi persona y en especial, a mi profesor y referente Carlos Bobed, por su seguimiento constante y productivas discusiones, correcciones y consejos sobre este TFG.

Capítulo 6

Bibliografía

[1] Franco Rezabek Steve Engledow Andrew Kane, Connor Kirkpatrick and Bob Strahan. Post call analytics for your contact center with Amazon language AI services, Dec 2021. Accedido por ultima vez 08-03-2023.

[2] Alex Chirayath Simran Baxendale and Shivani Mehendarge. Performing medical transcription analysis with Amazon Transcribe Medical and Amazon Comprehend Medical, May 2020. Accedido por ultima vez 08-03-2023.

[3] Julien Simon. Amazon Transcribe now supports automatic language identification, Sep 2020. Accedido por ultima vez 09-03-2023.

[4] Allen Guo, Arlo Faria, and Korbinian Riedhammer. Remeeting - Deep Insights to conversations. In *INTERSPEECH*, pages 1964–1965, 2016. Accedido por ultima vez 07-03-2023.

[5] Amazon Transcribe - developer guide, Nov 2017. Accedido por ultima vez 03-03-2023.

[6] Lak Sri Felipe Santiago, Pallavi Singh. *Building cognitive applications with IBM Watson services Vol. 6.* IBM Redbooks, May 2017. Accedido por ultima vez 27-02-2023.

[7] E. A. Epstein, M. I. Schor, B. S. Iyer, A. Lally, E. W. Brown, and J. Cwiklik. Making Watson fast. *IBM Journal of Research and Development*, 56(3.4):15:1–15:12, 2012. Accedido por ultima vez 01-03-2023.

[8] IBM Watson (Productor). IBM Watson: How it works, Nov 2014. Accedido por ultima vez 27-02-2023.

[9] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke. The Microsoft 2017 Conversational Speech Recognition System. 2017. Accedido por ultima vez 03-03-2023.

[10] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. Achieving human parity in conversational speech recognition, 2016. Accedido por ultima vez 02-03-2023.

[11] Binbin Xu, Chongyang Tao, Zidu Feng, Youssef Raqui, and Sylvie Ranwez. A benchmarking on cloud based Speech-To-Text services for french speech and background noise effect, 2021. Accedido por ultima vez 01-03-2023.

[12] Srikanth Machiraju and Ritesh Modi. Azure Cognitive Services. In *Developing Bots with Microsoft Bots Framework*, pages 233–260. Apress, December 2017. Accedido por ultima vez 07-03-2023.

[13] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision, 2022. Accedido por ultima vez 08-03-2023.

[14] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, 2020. Accedido por ultima vez 09-03-2023.

[15] Andrew C. Morris, Viktoria Maier, and Phil D. Green. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Interspeech*, 2004. Accedido por ultima vez 11-03-2023.

[16] Kesava Mandiga. Understanding Word Error Rate (WER) in Automatic Speech Recognition, Dec 2021. Accedido por ultima vez 14-03-2023.

[17] Ye-Yi Wang, A. Acero, and C. Chelba. Is Word Error Rate a good indicator for spoken language understanding accuracy. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*. IEEE. Accedido por ultima vez 16-03-2023.

[18] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. Automatic Speech Recognition Errors Detection and Correction: A Review. In *International Conference on Natural Language and Speech Processing*, 2015. Accedido por ultima vez 01-04-2023.

[19] John Wells. SAMPA - computer readable phonetic alphabet, Oct 2005. Accedido por ultima vez 03-04-2023.

[20] Correa Duarte and José Alejandro. Díptico de alfabetos fonéticos: Alfabeto Fonético Internacional (IPA), Alfabeto X- SAMPA y Alfabeto Fonético de la Revista de Filología Española (Contiene comandos para implementar el IPA en Praat). 2013. Accedido por ultima vez 03-04-2023.

[21] F. Schiel. Automatic Phonetic Transcription of Non-Prompted Speech. In *Proc. of the ICPhS*, pages 607–610, San Francisco, August 1999. Accedido por ultima vez 04-04-2023.

[22] Florian Schiel. A statistical model for predicting pronunciation. In *International Congress of Phonetic Sciences*, 2015. Accedido por ultima vez 04-04-2023.

[23] María G. Buey, Carlos Bobed, Jorge Gracia, and Eduardo Mena. A Domain Independent Semantic Measure for Keyword Sense Disambiguation. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 1883–1886, New York, NY, USA, 2021. Association for Computing Machinery. Accedido por ultima vez 11/05/23.

Lista de Figuras

Lista de Tablas

Anexos

Anexos A

Glosario de términos

A continuación se presentan algunos tecnicismos de la disciplina junto con su definición, con el fin de facilitar la lectura de los siguientes anexos.

- **Freestyle** Habilidad para rapear de forma improvisada.
- **MC** "Maestro de Ceremonias", rapero, persona que se dedica profesionalmente al *freestyle*.
- **DJ** Persona responsable de proporcionar las bases.
- **Host** "Anfitrión" de la batalla, la persona que presenta a los competidores y les ayuda a mantener un orden y fomentar el espectáculo. Su función es la de árbitro de la batalla.
- **Jurado** Conjunto de personas que deciden el resultado de una batalla.
- **Formato** Conjunto de reglas que fomentan la demostración de habilidad, ofreciendo estímulos y restricciones.
- **Base** "Beat" o canción sobre la que se hace rap.
- **Barra** División estructurada que existe en el rap, comparable a un verso de un poema.
- **Patrón** Conjunto de 4 barras, comparable a una estrofa de un poema.
- **Punchline** "Remate" de una rima, última barra que da sentido al patrón.
- **Flow** Ritmo y congruencia entre un rapero y la base. Se puede medir como la capacidad del artista de rapear a capella.
- **A capella** Hacer *freestyle* sin base.
- **Estructura** Orden y estructura de las barras y su rima dentro de un patrón.
- **Métrica** Relación del número de sílabas de cada barra en un patrón.
- **Técnica** Capacidad para generar estructuras y subestructuras (estructuras al nivel de barra, no de patrón) al hacer *freestyle*

- **Puesta en escena** Actitud en una rima, capacidad de expresar y transmitir emociones con esta
- **Réplica** Empate en una batalla de rap, representado con una equis con los brazos.

Anexos B

El dominio de las batallas de rap

Una batalla de gallos (o batalla de rap) es una competición en la que 2 o más MCs intentan medir quien es el mejor practicando freestyle.

Esta competición suele estar dividida en diferentes rondas, en las que los MCs se turnan rapeando con diferentes formatos.
Estos suelen hacerlo acompañados de un host, que ayuda al orden y el show, un DJ, que proporciona las bases y un jurado, que puntúan y deciden al ganador de la batalla.

B.1. Formatos

Tal como indica el glosario, el objetivo del formato es ayudar a los MCs a demostrar su habilidad, ofreciendo estímulos y restricciones. A continuación se detallan algunos formatos:

- **Minuto libre** El primer formato creado, se basa en 60" de ataque libre seguidos de 60" de respuesta del rival, con su posterior intercambio de roles. Esta explicación se puede observar en la Tabla B.1

MC 1	MC 2
60" ataque	
	60" respuesta
	60" ataque
60" respuesta	

Tabla B.1: Explicación formato "minuto libre"

- **4x4** Cada MC intercambia 4 barras (un patrón) con su rival 4 veces, formando un total de 32 barras (16 barras para cada uno). Esta explicación se puede observar en la Tabla B.2
  Existen variantes como el 2x2 o el 8x8, cambiando el número de barras de cada MC.

MC 1	MC 2
4 barras	
	4 barras
4 barras	
	4 barras
4 barras	
	4 barras
4 barras	
	4 barras

Tabla B.2: Explicación formato “4x4”

– **Easy/Hard Mode** Se basa en 60” en el que se proporciona una palabra aleatoria cada 10” o 5”. El MC debe usar esta palabra (o su campo semántico) en ese intervalo de tiempo. Posteriormente, se intercambian los roles con el rival. Esta explicación se puede observar en la Tabla B.3

MC 1	MC 2
60” con palabras cada 5-10”	
	60” con palabras cada 5-10”

Tabla B.3: Explicación formato “Easy/Hard Mode”

– **Deluxe** 3 patrones a capella por MC, seguido de 160” de 4x4. Esta explicación se puede observar en la Tabla B.4

– **Minuto a sangre** Análoga estructura al Minuto libre, cuyo único objetivo es atacar e insultar al rival. Se suelen usar los defectos físicos o errores de este.

– **Temática** Misma configuración que el Minuto libre, pero cada MC basa su minuto en la temática escogida aleatoriamente por la organización al inicio de la ronda.

MC 1	MC 2
1 patron a capella	
	1 patron a capella
1 patron a capella	
	1 patron a capella
1 patron a capella	
	1 patron a capella
∼160” de 4x4	∼160” de 4x4

Tabla B.4: Explicación formato “Deluxe”

– **Personajes contrapuestos** Idéntica forma al 4x4, donde se proporcionan diferentes personajes en los que el MC basa sus barras (Ej: Batman vs Superman)

– **Baúl de objetos** Mismo orden que el 4x4, pero en cada patrón, el MC sacará un objeto del baúl de objetos preparado por la organización, que usará para rapear.

– **Kickback** Misma disposición que el Minuto libre, pero en la primera barra de cada patrón el rival realiza una pregunta, que será contestada en los siguientes 3 patrones.

Respecto al análisis de la batalla, se puede relacionar algunos de estos formatos a un campo del NLP:

Formato	Posibles campos NLP
Easy/Hard mode	Detección de campos semánticos y familias léxicas
Temática, Baúl de objetos, Personajes contrapuestos	Detección de campos semánticos, entidades y propiedades, figuras literarias
Kickback	Extracción de consulta de lenguaje natural + extracción de respuesta a esta

B.2. Puntuación

Para decidir el ganador, los jueces califican cada ronda siguiendo un sistema de puntuación donde se valoran la coherencia, la contundencia y la prolijidad en la ejecución de las rimas, así como su adaptación al formato. Es importante que estos no valoren únicamente el punchline sino la rima al completo. Aunque cada juez tiene sus propias medidas, todos siguen una fórmula estandarizada de puntuación por ronda:

– Cada patrón es puntuado con un valor entre 0 y 4, con incrementos de 0.5 (Puntuaciones posibles: 0, 0.5, 1, (...), 3.5, 4) en base al punchline, al sentido del patrón, la métrica, la estructura, las figuras literarias, la originalidad, el seguimiento de la base, etc...

– Cada ronda es puntuado el flow, la puesta en escena y las técnicas con un valor entre 0 y 2 para cada una, con incrementos de 0.5 (Puntuaciones posibles: 0, 0.5, 1, 1.5, 2)

Un ejemplo de una plantilla de puntuación para el formato Deluxe podría ser la siguiente:

Figura B.1: Ejemplo de plantilla de votación formato Deluxe

El trabajo de los jueces es subjetivo, similar al análisis de la poesía, donde los gustos y preferencias influyen en la puntuación otorgada, así como las vivencias y el conocimiento del tema sobre el que se está rapeando. Estos deberían ser conscientes de estos sesgos e intentar corregirlos, acercándose a la objetividad y asegurando que el MC realmente está improvisando.

Tras sumar la puntuación de todas las rondas, se obtiene el resultado de la batalla. Si la puntuación de los dos MCs no dista más de 4 puntos se considera una réplica (empate), donde se repite parte del enfrentamiento y la votación.

B.3. Otras vertientes

Existen otros estilos y tipos de batalla. A continuación, se detallan los más comunes:

- **Batallas escritas** Modalidad donde los enfrentamientos se conocen con anterioridad y los MCs preparan sus rimas anteriormente por escrito (eliminando el carácter "improvisado" de las batallas).
- **Batallas de canciones** Modalidad donde el freestyle es reemplazado por canciones, normalmente escritas por el mismo MC, donde se valora su ejecución.
- **Batallas de calle** Batallas normalmente espontáneas, en parques o plazas, en las que no suele haber jurado. Es comparable a una batalla "amistosa".