UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

Aprendizaje semi-supervisado e interactivo para la anotación de un corpus de música histórica manuscrita

TRABAJO FIN DE MÁSTER

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Autor: Villarreal Ruiz, Manuel

Tutor: Sánchez Peiró, Joan Andreu

Curso 2020-2021

Resum

El reconeixement de música manuscrita és la tarea en la que s'empleen tecnologies de la computació per a, a partir d'una imatge d'un pentagrama musical, obtindre una transcripció. Si combinem açó amb técniques d'aprenentatge semi-supervisat, que permeten etiquetar grans conjunts d'informació a partir d'una petita part dels mateixos, i aprenentatge interactiu, que permet a un supervisor humà colaborar amb la màquina en el procés de transcripció, el que obtenim és un sistema que a partir d'unes poques transcripcions de música pot aconseguir una gran base de dades de gran qualitat.

Açò es important debut a que les bases de dades de música manuscrita estiquetades escassetgen, encara que la quantitat d'obres que és d'interés preservar es conten en l'ordre de millions, tasca inabordable per a les persones, pel que és necessari un mètode que ens permeta fer que totes aquestes obres sense etiquetar siguen etiquetades amb el menor esforç possible.

En aquest treball utilitzem tecnologies punteres del reconeixement de text manuscrit aplicades a la música, com són les xarxes neuronals, tant convolucionals com recurrents, i models de llenguatge per a explorar diferents métodes que faciliten l'etiquetat de conjunts de dades.

Utilitzem mesures com la probabilitat a posteriori i l'entropia de les mostres per a determinar com ha de distribuirse l'esforç humà al moment d'etiquetar mostres manualment, i mostrem diferents mètodes que determinen si una mostra etiquetada és o no adequada per a incluire's en el conjunt de dades aconseguint finalment un mètode eficaç per a anotar gran quantitats de mostres amb un esforç considerablement menor.

Paraules clau: música manuscrita, aprenentatge automàtic, xarxes neuronals, xarxes profundes, model òptic, aprenentatge semi-supervisat, aprenentatge interactiu

Resumen

El reconocimiento de música manuscrita es la tarea en la que se emplean tecnologías de la computación para, a partir de una imagen de un pentagrama musical, obtener una transcripción. Si combinamos esto con técnicas de aprendizaje semi supervisado, que permiten etiquetar grandes conjuntos de información a partir de una pequeña parte de los mismos, y aprendizaje interactivo, que permite a un supervisor humano colaborar con la máquina en el proceso de transcripción, lo que obtenemos es un sistema que a partir de unas pocas transcripciones de música puede conseguir una gran base de datos de mucha calidad.

Esto es importante debido a que las bases de datos de música manuscrita etiquetadas escasean, aunque la cantidad de obras que es de interés preservar se cuentan en el orden de los millones, tarea inabordable para las personas, por lo que se hace necesario un método que nos permita hacer que todas estas obras sin etiquetar sean etiquetadas con el menor esfuerzo posible.

En este trabajo utilizamos tecnologías punteras del reconocimiento de texto manuscrito aplicadas a la música, como son las redes neuronales, tanto convolucionales como recurrentes, y modelos de lenguaje para explorar distintos métodos que faciliten el etiquetado de estos conjuntos de datos.

Utilizamos medidas como la probabilidad a posteriori y la entropía de las muestras para determinar como debe distribuirse el esfuerzo humano a la hora de etiquetar muestras manualmente, y mostramos diferentes métodos que determinan si una muestra etiquetada es o no apta para incluirse en el conjunto de datos logrando finalmente un método eficaz para anotar grandes cantidades de muestras con un esfuerzo considerablemente menor.

Palabras clave: música manuscrita, aprendizaje automático, redes neuronales, redes profundas, modelo óptico, aprendizaje semi-supervisado, aprendizaje interactivo

Abstract

Handwritten music recognition is the task where computation technologies are used for, from an image of a musical score, obtaining a transcription. If we combine this with semi supervised learning techniques, that allow labeling big information sets from a small fragment of them, and interactive learning, that allow a human supervisor collaborating with the machine in the transcription process, we obtain a system that from a few music transcriptions can accomplish a big data set of great quality.

This is important given that labeled handwritten music data sets are scarce, even though the amount of pieces that is of interest to preserve are counted in the order of millions, an unapproachable task for people. This makes necessary a method that allows us to label this pieces with the least effort possible.

In this work we use the latest technologies for handwritten text recognition applied to music, such as neural networks, both convolutional and recurrent, and language models to explore different methods that make labeling easier for this data sets.

We use measures such as posterior probability and sample entropy to determine how should human effort be used when labeling samples manually, and we show different methods to determine if a sample is or isn't good enough to be included in the data set, accomplishing in the end an effective method to label big amounts of samples with a considerably lesser effort.

Key words: handwritten music, machine learning, neural networks, deep networks, optical model, semi-supervised learning, interactive learning

Índice general

Índice de figuras

CAPÍTULO 1
Introducción

Ottaviano Petrucci, fabricante de papel, impresor y editor italiano inventó el sistema de impresión de música por medio de tipos móviles. Con este sistema, en 1501 imprimió una colección de canciones polifónicas francoflamencas que fue la primera edición de música impresa de la historia [1].

Con este dato presente podemos ver como toda obra previa a esta fecha, 1501, es una obra manuscrita. La cantidad de obras que se tienen en este estado se cuentan en los millones, y estas obras no son accesibles en su mayoría, encontrándose únicamente disponibles mediante la lectura del documento original.

Todo esto resalta la importancia de investigar sistemas automáticos que permitan el reconocimiento, indexación y tratamiento de este tipo de obras. Sin embargo, este tipo de investigaciones parten siempre del mismo punto, al utilizar tecnologías dirigidas por los datos, como son las redes neuronales y las técnicas de aprendizaje automático, se necesitan bases de datos con las que trabajar.

En el campo del reconocimiento de la notación musical manuscrita o OMR (por su traducción al inglés Optical Music Recognition) nos encontramos con escasez de este tipo de bases de datos etiquetadas.

El enfoque de este trabajo será este punto clave, estudiar como podemos conseguir de manera eficiente aumentar la cantidad de datos etiquetados que se tienen para aplicar otras técnicas y tecnologías.

Para ello vamos a atacar este problema desde el punto de vista del aprendizaje semi-supervisado. El aprendizaje semi-supervisado se utiliza en el caso en el que, dada una gran base de datos donde solo una pequeña parte se tiene etiquetada, se utiliza esa parte para tratar de etiquetar el resto. Utilizaremos esto en conjunto con técnicas de aprendizaje activo o interactivo, que permiten a un usuario humano colaborar con el sistema de aprendizaje.

Vamos a estudiar distintas formas de determinar cuando una muestra que originalmente no estaba etiquetada y ahora sí lo está tiene una etiqueta válida, así como el mejor método para determinar sobre que muestras se debe realizar intervención humana para mejorar el sistema completo.

[1] https://es.wikipedia.org/wiki/Ottaviano_Petrucci

Además combinaremos esto con una nueva idea, introduciendo lo que llamaremos símbolo de error, permitiendo cierto margen de error en función de la confianza deseada, tratando de facilitar la tarea de la intervención humana en esta cuestión.

1.1 Motivación

Este trabajo tiene varias motivaciones principales, las cuáles pueden verse también en los objetivos.

La primera motivación es el querer trabajar con técnicas que necesitan de grandes bases de datos para poder aplicarse de manera consistente, por lo que es importante estudiar las maneras de conseguir las mismas en profundidad.

La segunda motivación, también de carácter técnico, es el estudio en si mismo, no sólo nos es de interés el resultado, por el que obtenemos una base de datos de calidad, sino que también es interesante el explorar otras técnicas utilizadas que forman parte del campo del reconocimiento de texto o música manuscritos.

La tercera y última motivación tiene un carácter distinto, no es estrictamente técnica, esta es el avance y la importancia de la conservación en el campo de la herencia cultural.

Esta es una tarea que hoy en día es abordable con las nuevas tecnologías y que antes era imposible de atacar, la herencia cultural a gran escala, y el poder aportar a este campo es una de las grandes motivaciones del trabajo.

Podemos ver como todas estas motivaciones en conjunto conforman un trabajo completo con una idea clara y bien definida.

1.2 Objetivos

Se presentan a continuación los objetivos de este trabajo, con sus objetivos secundarios asociados:

-Desarrollo e implementación de un sistema de aprendizaje semi-supervisado.

Esto incluye:

-Desarrollo e implementación de un sistema de reconocimiento de música manuscrita.

-Uso del sistema de obtención de grafos de palabras a partir de los resultados.

-Automatización del proceso completo.

-Desarrollo y uso de un sistema de aprendizaje interactivo.

Esto incluye:

-Estudio de distintos métodos de interacción humano-máquina, determinando cuando se debe actuar.

-Diseño de estos métodos de interacción.

-Exploración de ideas que faciliten el trabajo del usuario, mejorando la interacción de este con el sistema.

Para valorar si estos desarrollos se realizan de manera adecuada y si los resultados son sólidos y tangibles, planteamos:

-Realización de un experimento de referencia para comprobar que trabajamos sobre ideas sólidas.

-Realización de un distintos experimentos para probar todos los métodos establecidos y desarrollados.

-Análisis comparativo entre los distintos métodos puestos a prueba.

-Uso de técnicas de análisis adecuadas y medición de resultados de manera formal.

1.3 Metodología

Para lograr estos objetivos se va a proceder de la siguiente manera:

Lo primero que hemos hecho es utilizar las herramientas que tenemos al alcance de nuestra mano y nos son de confianza para repetir un experimento base sobre el corpus con el que vamos a trabajar, para asegurarnos de que los resutlados obtenidos van en la línea de lo que deberíamos esperar.

Para realizar este experimento se ha utilizado una herramienta de redes neuronales profundas, tanto convolucionales como recurrentes, para generar un modelo óptico, y modelos de lenguaje basados en modelos ocultos de Markov.

Una vez establecemos que estas herramientas van a ser adecuadas para la tarea a partir de ese experimento base procedemos a montar el sistema de aprendizaje semi-supervisado e interactivo, que nos permitirá explorar distintas opciones.

Este sistema itera sobre los datos mientras el sistema semi-supervisado realiza su función, añadiendo datos sin etiquetar al conjunto de los etiquetados. Cada cierto número de iteraciones entra en funcionamiento la parte interactiva, donde se determina de manera automática donde debe actuar el usuario humano y tras ello este interviene.

Debido a que tenemos que realizar una gran cantidad de experimentos esta última parte va a automatizarse, va a simularse esa interacción humana, y los pasos que haría un usuario se harán por el ordenador aunque se cuenten como interacción humana.

Finalmente, para el análisis de resultados de los experimentos realizados, utilizamos una serie de *scripts* que nos permiten evaluar distintas métricas sobre nuestros experimentos.

En este trabajo hacemos distintas medidas, por un lado tenemos que medir que el sistema sea capaz de hacer un buen reconocimiento, y que se mejoren los

resultados al obtener nuevos datos, por otro, hay que medir como de costosa es la intervención humana entre los distintos experimentos, con todo esto vemos como hay que ser rigurosos en este último apartado.

1.4 Estructura de la memória

En este primer capítulo hemos introducido el tema alrededor del que se centra el trabajo, hemos mostrado las motivaciones que hacen que este trabajo sea de nuestro interés así como los objetivos que nos permitirían satisfacer estas motivaciones y hemos explicado unas líneas generales de actuación que se seguirán a la hora de desarrollar el mismo.

En el siguiente capítulo, el segundo, hablaremos del contexto tecnológico que envuelve el sistema que se desea desarrollar, tanto para la parte del reconocimiento de música manuscrita como para la parte del aprendizaje semi-superivsado e interactivo.

Además, en este capítulo realizaremos una crítica a este contexto tecnológico, hablaremos de la importancia de las técnicas que conforman el mismo y haremos nuestras propias propuestas en cuanto a como se podría avanzar en el campo.

En el tercer capítulo, el asociado al marco de trabajo, en este hablaremos de los elementos que conforman el sistema base, sin entrar en el foco de la investigación, centrándose únicamente en los elementos base para el reconocimiento.

Con esto también hablaremos de la planificación del trabajo, de como se han preparado las distintas fases del proyecto, las actividades asociadas a cada una de estas fases y mostraremos un diagrama temporal con un reparto inicial de horas de trabajo.

En el cuarto capítulo entraremos en detalle en los distintos elementos que conforman nuestro sistema, especificando parámetros, tecnologías, herramientas e incluso el hardware utilizado a lo largo de nuestros experimentos, mostraremos también los datos a utilizar en esta experimentación.

Es aquí donde hablaremos más en detalle de los sistemas semi-supervisado e interactivo, veremos como se han conformado y bajo que premisas se utilizarán, en general, hablaremos de su diseño y de como se incorporan al resto de elementos del trabajo.

El quinto capítulo es en el que hablaremos de la experimentación, aquí se verán cada uno de los experimentos realizados, el protocolo de evaluación a seguir y como se ha realizado el proceso de entrenamiento de los distintos modelos a partir de nuestros datos.

Además de esto los resultados se mostrarán en este capítulo, y haremos cierto análisis sobre los mismos, tanto para el experimento base, como para cada uno de los experimentos sobre el aprendizaje semi-supervisado e interactivo.

En el capítulo de conclusiones, el sexto, hablaremos de como ha sucedido el proyecto, de las distintas conclusiones que se pueden extraer de los resultados y de si el procedimiento ha sido adecuado una vez hayamos visto el mismo al completo.

También relacionaremos las conclusiones y resultados obtenidos con los objetivos planteados al inicio del trabajo, con el fin de ver si hemos cumplido los mismos, así como su relación con los estudios cursados.

Finalmente, en el último capítulo, veremos como podría continuarse este trabajo y los diferentes elementos que hemos encontrado y podrían explorarse en el futuro.

CAPÍTULO 2
Contexto Tecnológico

En este trabajo el contexto tecnológico incluye dos áreas distintas, por un lado, la cuestión del reconocimiento de música manuscrita, que por su parte es interesante, y por otro lado, la cuestión del aprendizaje semi-supervisado, que se estudia más detenidamente en este trabajo.

Primero vamos a analizar el contexto en lo referido al reconocimiento de música manuscrita.

Partimos de un marco original, en el que las distintas partes del trabajo se dividían en fases. En estas fases se realizaban distintos procesos como separación de líneas y símbolos, segmentación de símbolos, ... [17]

Al dividir el problema en fases nos encontramos con una serie de problemas, primero, cada una de estas fases puede tener sus propios errores, lo que resulta en errores acumulativos en el proceso.

Además, algunas de las tareas que se realizan, como es la separación de líneas y símbolos, acaban teniendo más errores que ventajas aportan, su realización es por necesidad más que por beneficio real.

Posterior a este marco se introduce el uso de Modelos Ocultos de Markov [4], estos métodos holísticos consiguen mostrar que el camino de los métodos dirigidos por los datos pueden ser de gran utilidad en este campo.

Trabajos más recientes demuestran que algunas de esas fases originales deben evitarse, pues consiguen grandes resultados en cuanto al reconocimiento se refiere ignorando dichas fases como son por ejemplo la separación de líneas y símbolos [2].

Hoy en día, la clave en esta tarea es, como en otras tareas de reconocimiento de texto o tareas similares a la que aquí se trata, el uso de redes neuronales. En estas tareas similares se consiguen grandes resultados [3] y por ello mismo podemos ver como se han introducido estas técnicas, acompañadas de otros elementos como son los modelos de lenguaje [16, 23], al reconocimiento de música manuscrita.

Con esto establecemos de manera clara cuál es el estado de la tarea del reconocimiento de música manuscrita, que a nivel tecnológico está bastante establecida.

A continuación vamos a hablar del contexto que respecta al aprendizaje semi-supervisado.

La idea de utilizar tanto muestras etiquetadas como sin etiquetar en combinación o bajo un objetivo común tiene tiempo [7]. La relación entre las muestras etiquetadas y sin etiquetar es muy clara, y la capacidad que pueden tener los sistemas para etiquetar nuevas muestras es grande.

Estudios específicos en cuanto a la tarea del aprendizaje semi-supervisado [8] muestran como los sistemas no empeoran debido a que se añaden nuevas muestras, sino que esto puede ocurrir debido a un modelo incorrecto, no por el aumento de datos que originalmente no estaban etiquetados.

Esto lo que nos indica es que si el modelo es correcto, que es algo que habrá que evaluar en este trabajo, estas técnicas deberían conseguir cada vez mejores resultados.

Trabajos posteriores utilizan estas técnicas para distintos tipos de redes, en concreto en el caso de redes profundas, que es lo que a nosotros atañe [14]. Podemos ver como para este tipo de redes el acercamiento del aprendizaje semi-supervisado sigue funcionando adecuadamente.

Otros trabajos muestran como la combinación del aprendizaje semi-supervisado con el aprendizaje activo o interactivo da buenos resultados [6, 26]. Podemos ver como esta interacción entre usuario y máquina sumada al trabajo propio de la máquina es un buen método para combinar el uso de datos etiquetados con no etiquetados consiguiendo buenos resultados a varios niveles.

La combinación de algunas de estas ideas con las técnicas más punteras deberían mejorar el estado actual de la tarea. Así es como nosotros vamos a atacar la tarea que tenemos entre manos.

2.1 Crítica al contexto tecnológico

En lo que respecta al reconocimiento de música manuscrita el contexto tecnológico está relativamente establecido, siendo los últimos trabajos en las mismas tecnologías y consiguiendo buenos resultados.

Si bien esto es cierto, no quedan libres de pecado, pues hay elementos a atacar como pueden ser las combinaciones de obras con su letra o el tratar de unificar distintos sistemas de notación debido a que esta tarea está poco explorada en el campo, habiendo trabajos que hablan de ella sólo a partir de los últimos años.

En cuanto a lo que el aprendizaje semi-supervisado respecta, un elemento clave en el contexto tecnológico es la ausencia de trabajos en el campo del reconocimiento de música manuscrita, no encontramos artículos que traten de atacar la falta de datos en el campo de este modo.

Además vemos estos trabajos utilizan distintas técnicas y prueban diferentes métodos, pero incluso algunos de estos trabajos que son muy recientes no utilizan las tecnologías más novedosas, que debería ser un estándar entre ellos. En este trabajo vamos a plantear distintos métodos donde todos ellos permiten trabajar con este tipo de técnicas de última generación como son las redes neuronales.

Otro elemento que en cierta medida merece crítica es la dirección de los avances. En muchos de estos casos hay un usuario humano que debe de interactuar con el sistema, sin embargo, avances e investigaciones en el campo asumen a este usuario humano como un proceso, cuando esto no es realmente así.

Es necesario que estas investigaciones no miren solamente en pos de los resultados, en el sentido de obtener un mejor ratio de error o etiquetar una mayor cantidad de datos, sino también en facilitar el trabajo humano, reducir este esfuerzo. Por eso son importantes métricas que muestren resultados en función del esfuerzo humano.

2.2 La importancia del aprendizaje semi-supervisado e interactivo

Las técnicas que hoy en día están a la cabeza de la investigación en el reconocimiento de texto o música manuscritos tienen algo en común, todas son dirigidas por los datos.

Las redes neuronales, el aprendizaje profundo, todas estas tecnologías necesitan de grandes cantidades de datos, y cuantos más datos se tienen para entrenar mejores resultados se consiguen. Por esto vemos como el conseguir datos nuevos de manera eficiente es una parte necesaria para continuar avanzando con la investigación.

Una de las líneas que se estudia con frecuencia es el aumento de datos, con el que se busca, a partir de las muestras que ya se tienen, aplicar modificaciones y distorsiones para que el sistema pueda aprender los datos de manera más generalizada, pero esto no nos proporciona nuevos datos o situaciones reales, simplemente consigue sistemas más robustos, elemento positivo pero no clave en la cuestión.

No hay tantas maneras de conseguir nuevos datos, y estas pasan mayoritariamente por un experto que dedique su tiempo a etiquetar muestras, lo cuál es tremendamente costoso. Es clave buscar métodos que nos permitan conseguir nuevos datos, datos reales, que permitan expandir la capacidad que tenemos de aplicar técnicas y realizar investigaciones.

Este aprendizaje no es perfecto, no permite la generación de datos a partir de la nada, si se quiere etiquetar una colección nueva, con el fin de conseguir datos a partir de la misma, se necesita tener una parte etiquetada. En la figura 2.1 podemos ver la idea que sigue este tipo de aprendizaje, donde empezamos con una pequeña parte del corpus y vamos añadiendo iterativamente nuevas muestras.

Hay que destacar que el ritmo al que se pueden etiquetar nuevas muestras no es lineal, el sistema aprende de lo que puede ver, por lo que las muestras que tendrán suficiente confianza como para ser aceptadas serán aquellas que tengan similitud con las ya vistas.

Lo más normal es que en las primeras iteraciones de este proceso se consiga añadir un número muy notable de muestras que originalmente no estaban etiquetadas, y que la cantidad de estas disminuya progresivamente hasta que ya

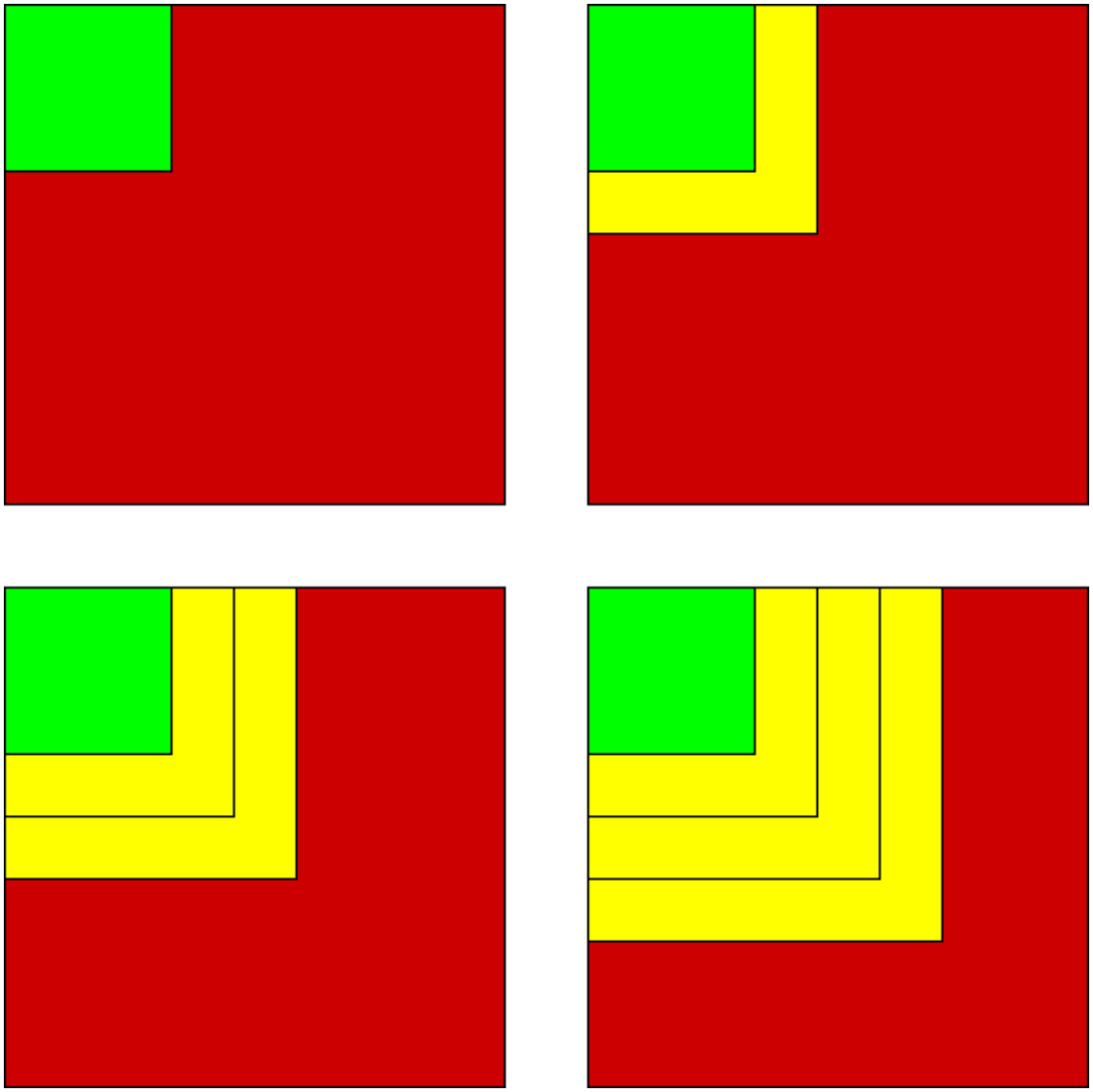

Figura 2.1: En verde datos anotados manualmente, en rojo datos sin anotar, en amarillo datos que se anotan automáticamente conforme el sistema progresa iterativamente.

no haya más muestras que anotar o, lo que es más común, el sistema no tenga suficiente información como para que las nuevas muestras superen el umbral de confianza.

Esta es la gran limitación que tiene el aprendizaje semi-supervisado, al utilizarse modelos dirigidos por los datos únicamente se puede aprender de aquello que encontramos en los mismos, por lo que eventualmente nos encontramos con muestras que nuestro sistema no es capaz de anotar y, por si mismo, no es capaz de conseguir nueva información que le permita salir de esta situación.

Es aquí donde entra la parte activa o interactiva, de alguna manera necesitamos conseguir muestras nuevas, que el sistema por si sólo no es capaz de obtener, para que pueda aprender a partir de estas y salir de su atasco. Para conseguir las muestras es necesario un usuario u operador humano que sea el que etiquete estas muestras, con ello el sistema puede continuar aprendiendo hasta detenerse nuevamente.

Es importante apoyar este trabajo, pues como ya hemos dicho, la intervención humana es lo más costoso de la tarea, por lo que queremos maximizar en la medida de lo posible la eficiencia y eficacia de estas anotaciones manuales.

Un elemento a determinar es que muestras debe anotar el usuario, y en este caso tenemos varias opciones:

- Muestreo aleatorio, se seleccionan muestras aleatorias con el fin de mantener uniformidad sobre el *corpus* al etiquetar.
- Selección manual, el usuario observa las muestras y decide cuáles debe anotar.
- Peores muestras, el sistema utiliza una métrica para determinar que muestras tienen menos confianza o se tiene mayor incertidumbre sobre las mismas y se etiquetan estas.

Elegir muestras aleatorias parece tener sentido en un principio, pues de esta forma no se introduce ningún tipo de sesgo en el sistema, sin embargo, estas muestras pueden no ser las que aporten una mayor información, y debido a que lo que queremos es que el sistema salga de un estado en el que se encuentra atascado, es más interesante buscar muestras que sean informativas.

En cuanto a medir como de informativas pueden ser las muestras puede hacerse de varias maneras, una es la segunda opción planteada, un experto humano que decide en función del sistema y las muestras que no superan la confianza cuáles considera que van a ser de mayor ayuda, sin embargo, esta no es una tarea fácil y se necesita mucho conocimiento para poder tomar este tipo de decisiones.

En definitiva, para elegir las muestras que aporten mayor información vamos a utilizar métodos similares a los que utilizamos para determinar la confianza de una muestra, o métodos basados en tecnologías de la información, como puede ser medir la entropía de las muestras con el fin de saber de que muestras tenemos una mayor desinformación.

Con esto deberíamos tener un sistema completamente funcional que sea capaz de, con el menor esfuerzo humano posible, anotar el mayor número posible de muestras.

2.3 Propuestas

En esta sección vamos a mostrar las propuestas que hacemos sobre lo planteado en el contexto tecnológico con el fin de obtener mejoras en la tarea y conseguir un buen sistema que sea capaz de etiquetar datos de manera semi-supervisada en el campo del reconocimiento de música manuscrita.

Por un lado el uso de este tipo de tecnologías en el campo de la música manuscrita ya es una propuesta en sí mismo, pues este campo está relativamente poco explorado y por ello no hay una gran abundancia de trabajos en el mismo.

En cuanto a la cuestión del reconocimiento de música manuscrita no vamos a hacer ninguna aportación, vamos a utilizar las técnicas utilizadas en los trabajos más recientes, sin hacer especial incisión en ninguna de las partes.

En cuanto a la cuestión del aprendizaje semi-supervisado, planteamos dos métodos para medir la confianza de las muestras, por un lado, vamos a utilizar la probabilidad a posteriori del resultado a partir del grafo de palabras como medida de confianza, cuando la probabilidad de una muestra sea lo bastante alta se aceptará la misma.

Esta medida plantea la necesidad de generar el grafo de palabras y en nuestro caso de utilizar un modelo de lenguaje, por ello, por otro lado, vamos a plantear una segunda medida de confianza, la confianza por replicación o votación, que es ajena al modelo utilizado.

En la confianza por replicación o votación se entrenan en cada iteración una serie de modelos, en lugar de únicamente uno, y si suficientes modelos se ponen de acuerdo en la transcripción de una muestra esta es aceptada como buena. Como esto trabaja sobre la transcripción final no es necesario un paso intermedio de cálculo de confianza.

En cuanto a la cuestión del aprendizaje activo o interactivo, específicamente en lo que respecta a la intervención humana, planteamos distintas propuestas con el fin de mejorar el sistema todo lo posible en este aspecto.

Primero planteamos dos métodos para seleccionar las muestras que debe el usuario humano etiquetar. El primero de estos dos métodos se hará a partir de la medida de confianza basada en la probabilidad a posteriori del grafo de palabras.

Lo que haremos es seleccionar las muestras con una menor probabilidad a posteriori, ya que entendemos que si el sistema asigna una probabilidad tan baja a estas muestras es porque carece del conocimiento para anotarlas, por lo que el etiquetado manual de las mismas debería aportar una gran cantidad de información.

El segundo de estos métodos se hará a partir de una medida de información propiamente dicha, la entropía derivacional de las muestras sin etiquetar. La entropía de una muestra es mayor cuanto mayor es la desinformación de la misma, a mayor falta de información, mayor entropía.

Elegiremos las muestras con una mayor entropía pues estas son las muestras que tienen una mayor desinformación y esto es exactamente lo que queremos resolver, la falta de conocimiento por parte de nuestro modelo.

Segundo, introducimos lo que llamaremos símbolo de error, que nos dará flexibilidad para trabajar con las muestras. Este símbolo funciona de la siguiente manera:

La manera de aceptar una muestra por su nivel de confianza debe cumplir dos condiciones, por un lado, la probabilidad de la muestra completa debe superar el umbral de confianza, por otro, cada uno de los símbolos que conforman la muestra deben haberse reconocido con una probabilidad que supere ese mismo umbral de confianza.

Esto lo que nos lleva es a muestras que tienen una gran confianza, superando el umbral, pero en las que alguno de sus símbolos no superan ese umbral, por lo que la muestra, aunque únicamente hay uno o dos símbolos que realmente han fallado o no deberían considerarse, se descarta.

Lo que haremos es sustituir estos símbolos donde la confianza es demasiado baja por el llamado símbolo de error, y permitiremos que las muestras que tengan menos de cierto porcentaje de errores pasen a formar parte de nuestro *corpus*.

Podemos ver como la introducción de un carácter erróneo no nos acaba proporcionando muestras correctas, por lo que tras cada iteración de nuestro sistema, una vez el usuario humano debe etiquetar nuevas muestras para aportar información, deberá además corregir los símbolos de error.

Buscamos con esto poder añadir muestras que originalmente no serían aceptables por la confianza de algunos de sus símbolos necesitando corregir únicamente los errores puntuales, ya que sabemos que el resto de la muestra sí consigue el nivel de confianza adecuado.

Con todo esto pretendemos no sólo poner a prueba las técnicas del aprendizaje semi-supervisado en el campo del reconocimiento de música sino que además planteamos ideas innovadoras con el fin de mejorar el trabajo que se puede hacer en el mismo.

CAPÍTULO 3
Marco de Trabajo

En este capítulo vamos a ver como deben construirse cada uno de los elementos que conforman el sistema que utilizamos, para poder entender el funcionamiento del mismo.

Aquí explicaremos qué es cada una de las partes aunque no necesariamente veremos las características específicas del sistema, sino que estableceremos el funcionamiento general del mismo.

3.1 Sistema de reconocimiento

3.1.1. Modelo Óptico

La primera parte de este sistema de reconocimiento, el modelo óptico, es la que permite a partir de una imagen obtener una serie de probabilidades asociadas a los símbolos que podemos encontrar en ella.

Los sistemas más utilizados para esto son las redes neuronales. En este caso se combinan redes convolucionales y redes recurrentes. Las redes convolucionales se encargan de la parte de extracción de características a partir de la imagen, las redes recurrentes utilizan estas características para obtener una serie de símbolos que conforman la transcripción.

En la figura 3.1 podemos ver como se conforma una red de este tipo, denominada CRNN.

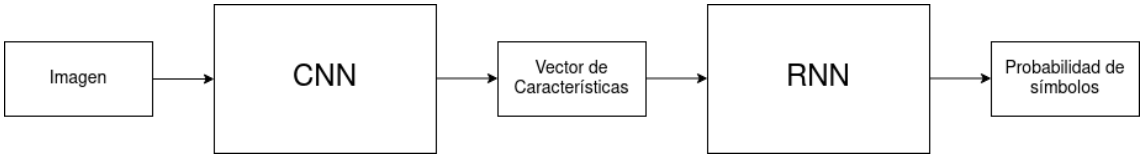

Figura 3.1: Estructura general del modelo óptico.

Este tipo de modelo se conoce como modelo de imagen a secuencia, y son utilizados comunmente en estas tareas de transcripción, ya sea transcripción de música, texto, ...

Capas convolucionales

Las capas convolucionales son excelentes en la extracción de características apartir de imágenes, se benefician especialmente de patrones jerárquicos en estas, pudiendo aprender patrones complejos a partir de otros más sencillos.

Para extraer este conjunto de características las capas aplican el operador de convolución sobre la imagen, la capa de entrada realiza un barrido sobre la imagen aplicando esta operación sobre la misma y obteniendo como salida una serie de vectores con mayor profundidad que la imagen original y generalmente con tamaño reducido.

Las subsecuentes capas convolucionales aplicarán estos mismos operadores de convolución sobre la salida de la capa anterior, con esto se van concentrando los cúmulos de información y se realiza una extracción adecuada de características.

Estos vectores de características se encuentran en la forma de un tensor, que es la estructura de datos con la que trabajan este tipo de redes neuronales.

Hay otros factores que afectan a la parte convolucional del modelo, como son las operaciones de *pooling*, *down-sampling* y la función de activación de las mismas, pero debido a que esto puede variar notablemente de una red a otra hablaremos de las características específicas de nuestra red en la fase de diseño.

Capas recurrentes

Las capas recurrentes estudian la información a lo largo del timepo, con ellas buscamos conseguir representar la conexión temporal entre los datos y obener una probabilidad de símbolos a partir de las características de la imagen.

En el caso del reconocimiento de música o texto el factor tiempo viene dado por los *frames* de la imagen, conforme se avanza según el orden de lectura (en nuestro caso de izquierda a derecha) se considera que avanza este "tiempo".

Las capas recurrentes se componen de unidades LSTM, estas resuelven el problema del desvanecimiento del gradiente, un problema conocido de las capas recurrentes más clásicas [9].

En la figura 3.2 podemos ver una representación de este problema del desvanecimiento del gradiente en función del tiempo.

No se va a explicar en detalle el funcionamiento de las unidades LSTM, pues es una tarea extensa, nos limitamos a remitir al lector al lugar en el que encontrar esta información [13].

De la misma manera que en el caso de las capas convolucionales, la salida de una capa recurrente es la entrada de la siguiente.

Lo que consiguen estas capas recurrentes es conectar las capas convolucionales y las características que se obtienen a través de estas con la salida 'real' o esperada.

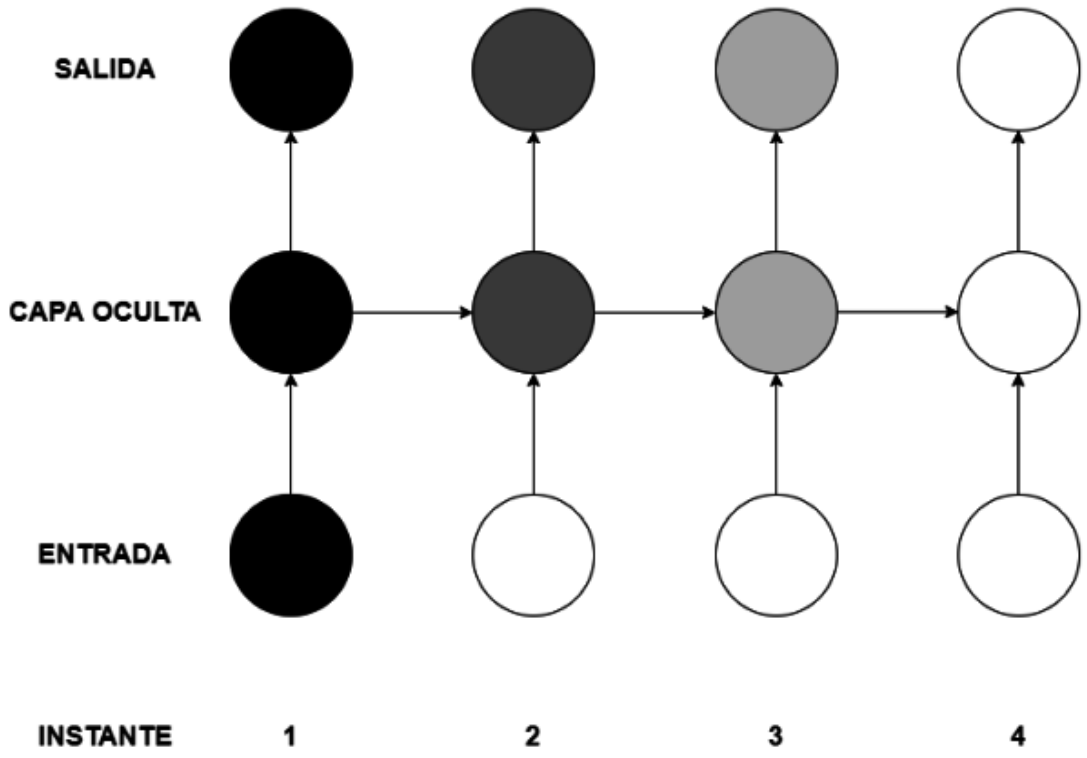

Figura 3.2: Representación del desvanecimiento por gradiente.

3.1.2. Modelo de Lenguaje

El modelo de lenguaje tiene como objetivo introducir al reconocimiento información sintáctica, a partir de una probabilidad externa al modelo óptico se introducen las características basadas en el lenguaje extraídas de los datos de entrenamiento.

La decodificación a partir únicamente del modelo óptico se hace a dada una imagen x y se obtiene la secuencia de mayor probabilidad $\hat{s}$ a partir de las distintas secuencias que puede deducir el modelo óptico, como podemos observar en la ecuación 3.1.

$$\hat{s} = arg \max_{s} P(s|x) \tag{3.1}$$

Para hacer uso del modelo de lenguaje introducimos a la ecuación 3.1 la probabilidad obtenida por el modelo de lenguaje o probabilidad a priori $P(s)$ que depende únicamente de la secuencia y sus características sintácticas, y no de la imagen. En este caso la ecuación a resolver en el proceso de codificación es la obtenida en la ecuación 3.2.

$$\hat{s} = arg \max_{s} P(s|x) = arg \max_{s} P(s)\, P(x|s) \tag{3.2}$$

El modelo de lenguaje utilizado es el modelo típico utilizado en estas tareas de reconocimiento de texto manuscrito o música manuscrita, este es, un modelo de n-grmas.

En un modelo de n-gramas se busca utilizar el conocimiento previo de la secuencia para tratar de predecir cuál es el siguiente elemento en la misma. El valor de n determina cuantas palabras se tendrán en cuenta previas a la que se quiere reconocer.

En las ecuaciones a continuación vamos a ver como se aproxima la probabilidad de un elemento w de la secuencia:

$$P(w) \approx \prod_{i=1}^{m} P(w|\Phi(w_1 \dots w_{i-1})) = \prod_{i=1}^{m} P(w|w_{i-n+1} \dots w_{i-1}) \tag{3.3}$$

Para estimar $P(w|w_{i-n+1} \dots w_{i-1})$ se utilizan las cuentas de frecuencias relativas $f(.|.)$:

$$P(w|w_{i-n+1} \dots w_{i-1}) = f(w|w_{i-n+1} \dots w_{i-1}) = \frac{C(w|w_{i-n+1} \dots w_{i-1} w_i)}{C(w|w_{i-n+1} \dots w_{i-1})} \tag{3.4}$$

Con esto definimos el clásico modelo de n-gramas utilizado en este tipo de sistemas. Si bien se definen otros elementos lingüísticos en trabajos recientes en cuanto al reconocimiento de música manuscrita se refiere, no son aplicables a nuestros datos, pues dependen de la doble representación de las características musicales y nuestro *corpus* únicamente tiene características singulares.

3.2 Entrenamiento

Para entrenar la parte convolucional del sistema se utiliza el algoritmo de *Back Propagation* [12], este es un conocido algoritmo de retro propagación del error, donde se compara la salida del sistema con la etiqueta de las muestras y se transmite el error por el sistema.

De esta manera se optimiza mediante descenso por gradiente, un proceso de optimización iterativa, buscando el mínimo en una función diferencial. En este caso utilizamos la función de activación *LeakyReLu* que veremos con más detalle en la parte de diseño.

Para entrenar la parte recurrente de nuestro modelo óptico se utiliza el algoritmo de *Back Propagation Through Time* [24], que es una versión del algoritmo de *Back Propagation* que tiene en cuenta el instante de tiempo en el que se hace el entrenamiento.

El último elemento importante dentro del entrenamiento es el algoritmo CTC [10], este algoritmo hace un alineamiento entre los *frames* de la imagen y la secuencia que queremos obtener. Esto es necesario ya que, de manera natural, una imagen no presenta una secuencia temporal, y queremos entender el avance de estos *frames* en el orden de lectura (aquí de izquierda a derecha) como nuestro avance del tiempo.

3.3 Planificación y estructura de trabajo

En este capítulo hemos hablado del marco de trabajo desde un punto de vista técnico, pero también hay que ver como se comprende el marco desde el punto de vista organizativo. Hay que determinar como se debe proceder para asegurarnos de que el proyecto se lleva a cabo de manera adecuada.

Esto pasa por determinar cuáles son las fases en las que se ha desarrollado el proyecto, las distintas actividades que deben realizarse en estas fases y determinar el tiempo necesario para llevar a cabo las mismas, haciendo una diferencia entre horas hombre y horas totales del proyecto.

3.3.1. Fases del proyecto

Vamos a establecer aquí las fases en las que se divide el proyecto, hay que tener en cuenta que al ser un trabajo de investigación, un trabajo de experimentación, las pautas a seguir van a diferenciarse de las más típicas en el campo empresarial o de la informática.

En la primera fase debemos determinar de manera concreta y concisa cuál es el problema que se va a tratar en el trabajo. Con esto buscamos dejar claros los caminos que podemos tratar de explorar y cuáles no llegarán a funcionar dado lo que buscamos.

Es importante en esta fase delimitar el problema, establecer los límites a los que estamos dispuestos a llegar a nivel de trabajo a realizar, determinar los recursos de los que disponemos para poder desarrollar el mismo y tener claro aquello que queremos tratar, para evitar cambios de rumbo en el desarrollo.

La segunda fase es una fase de establecimiento de la solución o lluvia de ideas, aquí pretendemos determinar cuáles son las opciones que tenemos en cuanto a la investigación en base al problema que hemos establecido previamente.

Aún con el nombre de la fase, no es estrictamente necesario definir la solución al completo, eso será trabajo de la fase de diseño, pero sí debemos dejar claro como vamos a actuar, cuáles son las posibles ideas a desarrollar que nos ayudarán a resolver el problema que tratamos, haciendo esto de la manera más concisa posible para facilitar el trabajo de futuras fases.

Estas dos fases comprenden lo que sería el estudio inicial, que se refleja en los dos primeros capítulos de esta memoria, donde se muestra el problema que tenemos que tratar y el contexto sobre el que trabajamos para llegar a resolver este.

La fase de diseño y desarrollo es el momento de detallar y estudiar con detenimiento lo planteado en la fase de establecimiento de la solución. Esto incluye el diseño de cada una de las partes del sistema, desde los distintos modelos hasta las métricas o sistemas de evaluación.

Además, no sólo hay que plantear como serán las piezas del trabajo sino que hay que darles forma, con esto nos referimos a que, si bien la mayoría de herramientas que se usan son herramientas ya implementadas, hay que combinar

las mismas y construir un sistema completamente funcional y eficiente para un correcto desarrollo.

La fase de experimentación, como su propio nombre indica, es la fase en la que se realizan los experimentos que se han planteado y a partir de lo desarrollado en la fase anterior.

Esta fase no requiere de tanto trabajo humano como puede ser la fase anterior, debido a que la mayoría de estos experimentos no requieren una interacción constante del usuario con los mismos para poder desarrollarse.

Sin embargo, esto no hace que las horas hombre asociadas a los mismos sean nulas, pues el mantenimiento de los sistemas en los que se desarrollan estos experimentos así como la revisión de que no sucede ningún problema con los mismos sí requiere esfuerzo humano.

La quinta fase de este proceso es una fase de análisis de resultados. Esta fase es especialmente importante en un proyecto de investigación, de hecho, esta fase es el punto clave que da validez y verosimilitud al resto del proyecto.

Tenemos que poder ver y demostrar que los resultados que se han obtenido son adecuados, y poder defender que las decisiones que hemos tomado son las que nos han llevado a estos mismos, así como poder identificar que cosas han fallado o donde nos hemos podido equivocar si los resultados fueran negativos.

La última y sexta fase es la que comprende la tarea de redacción o generación de un reporte donde se describa el trabajo realizado y se muestren los resultados obtenidos. La divulgación científica siempre debería formar parte de un proyecto de investigación, y con una memoria se cubre este ámbito.

3.3.2. Paquetes de actividades

En la figura 3.3 se puede observar un diagrama donde se despliegan los distintos paquetes de actividades asociados a cada una de las fases del proyecto. Estas deberán completarse para dar por superadas estas fases.

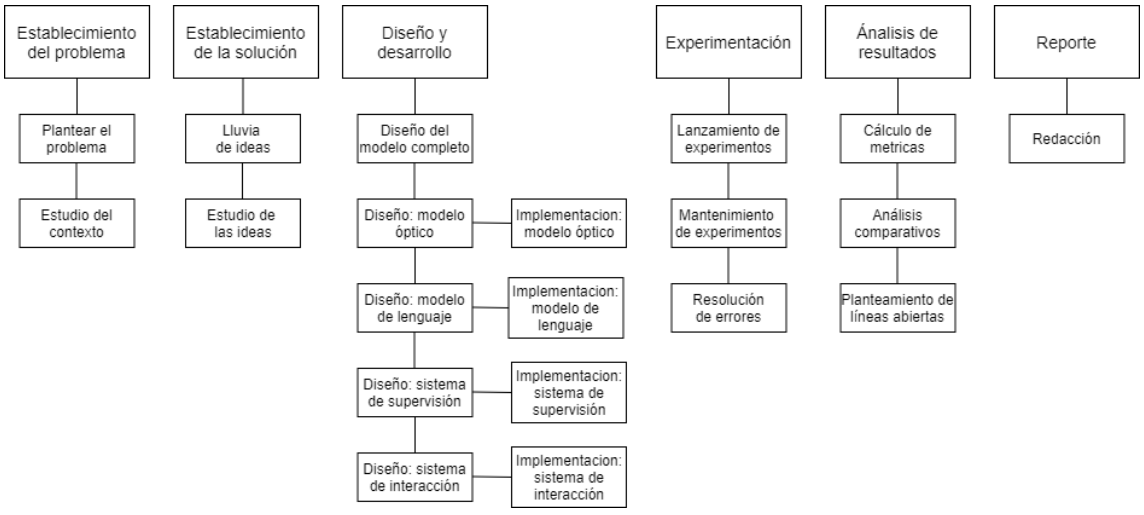

Figura 3.3: Paquetes de actividades asociadas a cada fase del proyecto de investigación.

A continuación vamos a describir con más detalle los distintos paquetes de actividades asociados a estas fases, desarrollando que acciones implica cada uno. Empezando por la primera fase, tenemos los siguientes paquetes de actividades:

- Planteamiento del problema, donde se establece el problema que vamos a tratar, en este caso la falta de datos y como solucionar la misma, así como las limitaciones de este.
- Estudio del contexto, donde se estudian las distintas tecnologías que han atacado este problema o problemas parecidos con el fin de dar una solución.

En la segunda fase del proyecto nos encontramos de nuevo con dos paquetes de actividades a realizar para dar por completada la misma, estas son:

- Lluvia de ideas, donde se plantearán diversas posibles soluciones e ideas que nos permitan atacar el problema que hemos planteado en la primera fase.
- Estudio de las ideas, donde se verán con detenimiento estas ideas planteadas, se estudiará la factibilidad de las mismas, y se cribarán aquellas mejores ideas para su diseño e implementación.

Estos dos conjuntos de actividades suponen el fundamento del trabajo, una correcta identificación del problema y de como se debe proceder para solucionarlo es la clave para que el resultado del trabajo sea satisfactorio, y por ello tienen gran importancia.

La tercera fase es la que tiene un mayor número de paquetes de actividades, y es también la que requiere un mayor número de horas hombre, un mayor esfuerzo humano, en su desarrollo. A continuación vemos las actividades que la componen:

- Diseño del modelo completo, donde se establece como debe funcionar el sistema en conjunto, entendiendo cada una de las piezas como una caja negra que se combinará con el resto.
- Diseño e implementación del modelo óptico, donde se establece el modelo a utilizar y se incorpora este al sistema completo de aprendizaje semisupervisado.
- Diseño e implementación del modelo de lenguaje, donde se establece como funcionará el modelo de lenguaje y se incorpora este al sistema de reconocimiento en conjunto al modelo óptico.
- Diseño e implementación de la supervisión del aprendizaje, donde se estudian los diferentes métodos para realizar el proceso de supervisión y se implementan estos mismos en conjunto con el resto de elementos.
- Diseño e implementación del sistema de interacción, donde se establece como va a realizarse el proceso de interacción usuario máquina y se introduce en el sistema.

A continuación vamos a ver las actividades de la cuarta fase, esta fase es una de las que requiere menos trabajo a nivel de horas hombre, sin embargo, es la fase donde vamos a encontrarnos con un mayor número de horas de cómputo pues es la de experimentación.

- Lanzamiento de experimentos, donde se pondrán en marcha los distintos experimentos, distribuyéndose en la red de trabajo que sea disponible y poniéndose los sistemas en funcionamiento.

- Mantenimiento de experimentos, donde, debido a que los experimentos que se realizan tienen una larga duración, se deberá hacer una monitorización periódica con el fin de asegurarse de que todo funciona adecuadamente.

- Resolución de errores, donde a lo largo de la experimentación se tratarán de solventar todos los errores que puedan aparecer y no hayan sido previstos en la fase anterior.

Ahora nos encontramos con, como ya hemos dicho, una de las fases más importantes del proyecto, pues un buen análisis es esencial. A continuación presentamos las actividades que hay que completar para satisfacer los objetivos relacionados con la misma:

- Cálculo de métricas, donde se utilizarán diversos *scripts* que nos permiten obtener mediciones sobre los resultados para poder posteriormente comparar experimentos.

- Análisis comparativos, donde se compararán los resultados obtenidos a partir de las métricas entre los distintos experimentos realizados con el fin de poder sacar conclusiones.

- Planteamiento de líneas abiertas, donde se describirán y estudiarán ligeramente las distintas líneas donde el trabajo podría continuar que se han encontrado al realizar el trabajo.

Por último, pero no menos importante, tenemos la fase final, donde se generará el documento o reporte que contiene los resultados del proyecto y el historial de como se ha realizado el mismo:

- Realización de la redacción, donde se plasmarán los resultados sobre el papel así como se contará de manera detallada todo el proceso.

3.3.3. Diagrama de Gantt y análisis de tiempo

En este apartado vamos a hablar del reparto temporal, de las horas que, a priori, se necesitan para la realización de este proyecto. Esto va a considerar las clásicas horas hombre, que son las horas de trabajo necesarias por parte una persona, pero no las horas de cómputo, que son las horas de trabajo computacional necesarias.

En la figura 3.4 podemos ver el diagrama de Gantt resultante de la organización del proyecto. Como podemos ver, el reparto de horas sigue una organización razonable, encontrando un mayor número de horas repartidas en aquellas fases donde se espera más trabajo.

Hay que tener en cuenta la ya mencionada diferencia entre horas hombre y horas de cómputo de cara a considerar el coste de un proyecto como este, las horas de cómputo son más baratas aunque no deben obviarse.

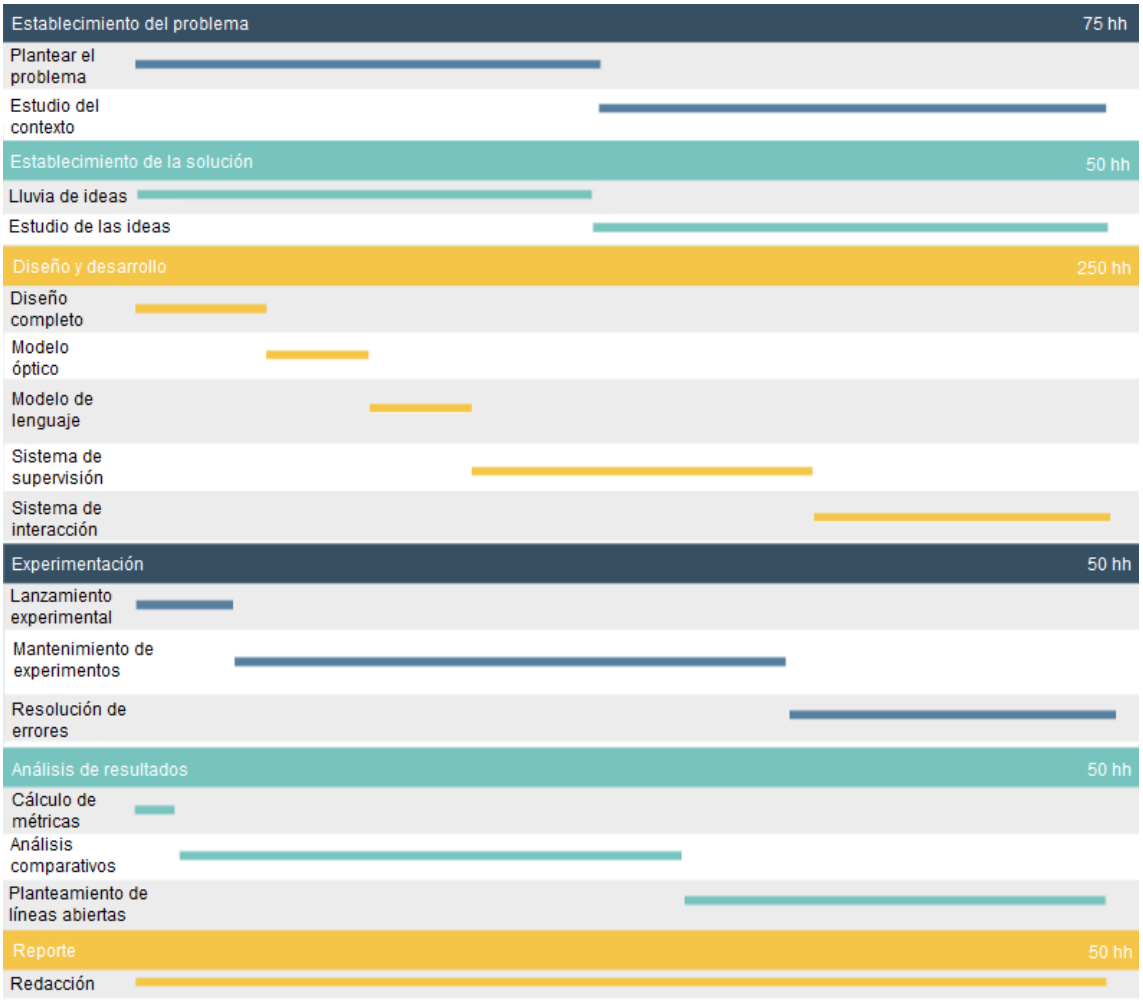

Figura 3.4: Diagrama de Gantt con las horas hombre asociadas al proyecto.

CAPÍTULO 4
Diseño

4.1 Diseño del sistema de reconocimiento

El sistema de reconocimiento consta de dos modelos que se entrenan a partir de una serie de imágenes de entrada ya etiquetadas, estos modelos son el modelo óptico y el modelo de lenguaje, que obtendrán las transcripciones.

A continuación vamos a ver en detalle cada uno de estos dos modelos, como hemos compuesto los mismos, los criterios de selección de parámetros y las distintas decisiones tomadas en su construcción.

4.1.1. Diseño del modelo óptico

En trabajos recientes se establece como debe formarse un buen modelo óptico basado en CRNN [3], pues estos son los modelos que mejores resultados han dado en el campo. Es por esto que nosotros vamos a basarnos en estos modelos para determinar como se construye el nuestro.

Hay distintas cuestiones que se consideran en estos trabajos previos, como son la normalización de las imágenes corrigiendo la torcedura y el color, sin embargo, los más recientes muestran como las redes neuronales son capaces de, con mínimos retoques a las imágenes, conseguir muy buenos resultados [23].

En la tabla 4.1.1 a continuación podemos ver los parámetros del modelo.

Capa	Características
Convolucional (16)	*Max-pooling* (2)
Convolucional (24)	*Max-pooling* (2)
Convolucional (48)	*Max-pooling* (1)
Convolucional (96)	*Max-pooling* (1)
Rnn LSTM (256)	*Dropout* (0.5)
Rnn LSTM (256)	*Dropout* (0.5)
Rnn LSTM (256)	*Dropout* (0.5)
Salida	*Lin. Dropout* (0.5)

Tabla 4.1: Parámetros utilizados en el modelo óptico.

Es necesario hablar de algunos de los elementos que ayudan a formar el modelo, como son la función de activación utilizada y las distintas operaciones de *pooling*, en este caso el ya mencionado *max-pooling* así como *adaptive pooling*. A continuación veremos los mismos.

Función de activación

Se necesita una función de activación para las capas convolucionales, generalmente se utilizan funciones de activación de la familia de la función *softmax*, en nuestro caso utilizaremos la función *LeakyReLu*, que es una variación de la función *ReLu* [25]. La función *ReLu* es de la siguiente manera:

$$f(x) = max(0, x) \tag{4.1}$$

Esta función tiene un pequeño problema de pérdida de información, pues si el valor de x es negativo, independientemente del valor absoluto del mismo, se pone a 0 y no se obtiene nada de ese estado.

La función *LeakyReLu* introduce un escalado a la parte negativa, para reducir en gran medida su valor, pero no ignorarlo (en la figura 4.1 podemos ver gráficamente como se representa la ecuación), como se muestra a continuación:

$$f(x) = max(0{,}1x, x) \tag{4.2}$$

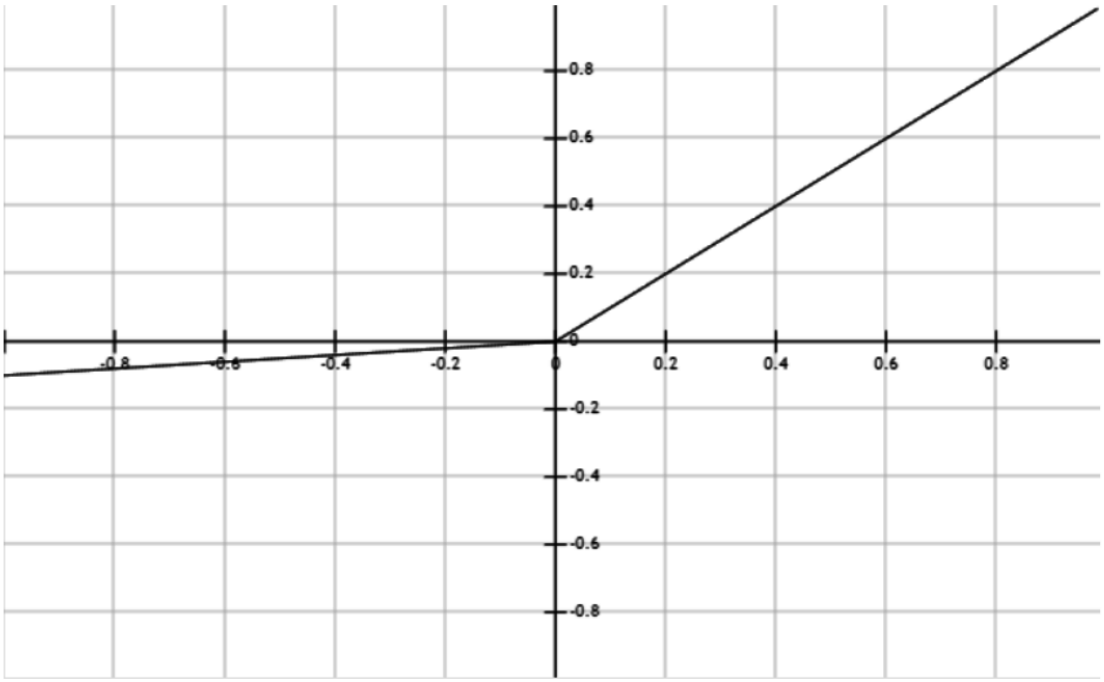

Figura 4.1: Forma que toma la función de activación *LeakyReLu*

Operaciones de *pooling*

En lo que respecta a las operaciones de *pooling* en este caso utilizamos dos tipos de operaciones con diferentes objetivos, la primera, la operación de *max-pooling*, es una operación de extracción de características, que trata de centrarse en las características que más destacan de entre un conjunto, quedándose la de mayor valor. En la figura 4.2 podemos ver un ejemplo de como funciona esta.

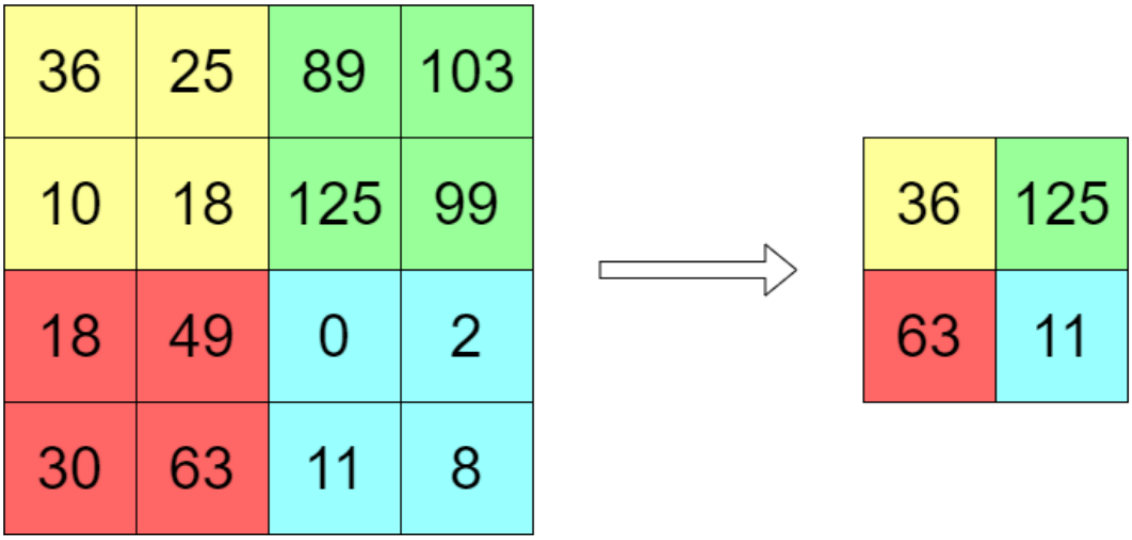

Figura 4.2: Ejemplo de una operación de *Max-pooling*

La segunda operación de *pooling* que utilizamos es el *adaptive pooling*, esto tiene la función de adaptar el tamaño de las imágenes de entrada con el fin de aprovechar todas las características posibles de las mismas, sin tener que reducir su tamaño de manera externa.

Como se puede ver en trabajos previos a mayor altura de las imágenes de entrada los resultados son mejores [3], esto es, cuanto más parecidas sean a la imagen original, mejores los resultados, por lo que tratar de mantener las imágenes todo lo parecidas a la original es lo ideal.

Además, en el campo del reconocimiento de música manuscrita ya se ha utilizado este tipo de técnicas para evitar la normalización del tamaño en las imágenes de entrada [23], por lo que podemos asumir que su uso está justificado respecto a tener que aplicar otras técnicas de reducción del tamaño de la imagen.

4.1.2. Diseño del modelo de lenguaje

En este apartado vamos a ver en algo más de detalle el modelo de lenguaje utilizado, aunque ya hemos descrito que se trata de un modelo de n-gramas en el capítulo anterior, y hemos visto como funcionan estos, todavía hay detalles a concretar.

Primero, los modelos de n-gramas dependen del parámetro n y se ven enormemente afectados por este, por lo que determinar el valor del mismo es muy importante.

Para establecer el valor de este parámetro debemos hacer pruebas con distintos valores y quedarnos el mejor en función del SER (*diplomatic symbol error rate*). Para hacer este estudio utilizaremos el experimento base, cuyos resultados se mostrarán en la parte de experimentación, y compararemos valores de n desde 2 hasta 5, para ver si hay una clara tendencia.

El segundo elemento que hay que considerar es el caso de los posibles descuentos a utilizar. Un descuento se encarga de asignar algún tipo de probabilidad a aquellos elementos que no se hayan visto en el entrenamiento del modelo, por lo que es esencial para evitar secuencias de probabilidad cero.

Vamos a probar dos tipos de descuento para cada uno de los tamaños de n, con el objetivo de encontrar el que obtenga mejores resultados, este estudio también se realizará sobre el experimento base. Los descuentos a probar son el descuento Knesser-Nay [11, 18, 21] y el descuento Witten-Bell [11, 21, 22].

Se observa que el uso de un valor de n igual a 2 utilizando el descuento de Witten-Bell obtiene los mejores resultados por lo que estos serán los parámetros a utilizar en nuestros modelos de lenguaje.

4.2 Aprendizaje semi-supervisado

En esta sección vamos a definir como funciona el sistema de aprendizaje semi-supervisado, vamos a ver el flujo de trabajo que debe seguirse y hablaremos de los distintos métodos y técnicas aplicados en detalle.

Lo primero que hay que mostrar es como va a funcionar el sistema. En la figura 4.3 podemos ver como es el flujo para el sistema sin margen de error, donde las líneas deben superar el umbral de confianza como líneas completas y para cada uno de sus símbolos.

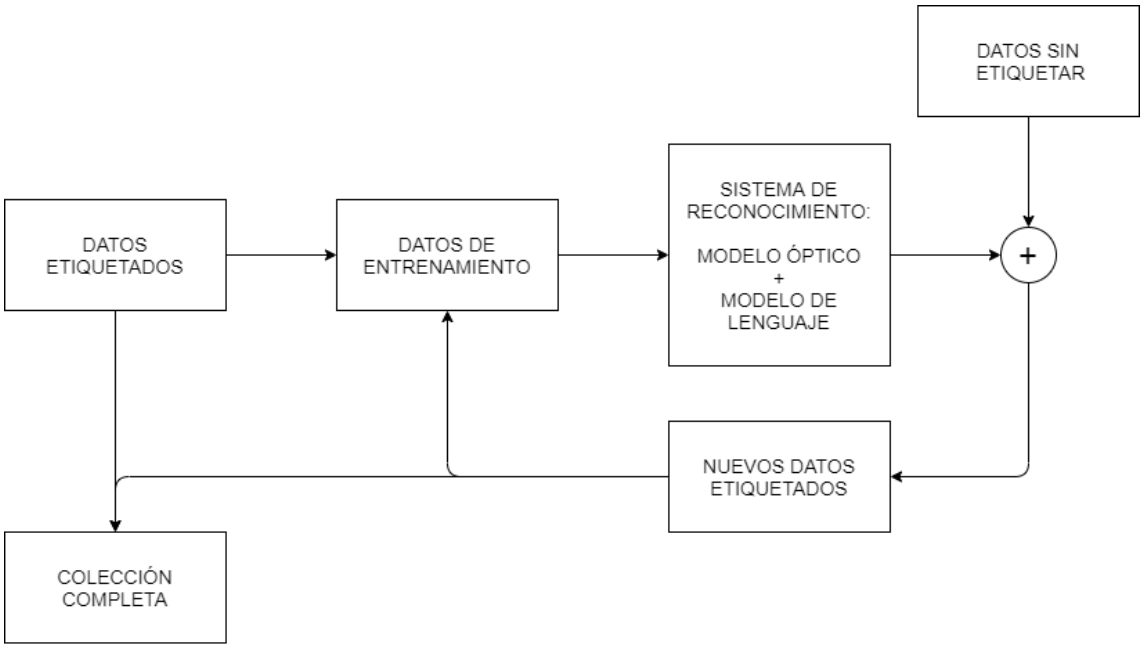

Figura 4.3: Esquema de funcionamiento para aprendizaje semi-supervisado sin margen de error.

Podemos ver en la imagen como los datos de entrenamiento para cada iteración se formarán a partir de los datos etiquetados originalmente y de los nuevos datos etiquetados y que han superado el umbral de confianza en la iteración anterior. Para la primera iteración se utilizarán únicamente los datos originales como entrenamiento.

Para generar los nuevos datos etiquetados se entrenará el modelo a partir de los datos de entrenamiento y tras esto se utilizará ese mismo modelo para reconocer las imágenes que están sin etiquetar.

Una vez hayamos obtenido estos datos utilizaremos los distintos métodos de confianza para saber si son adecuados, y, si lo son, se añadirán tanto a los datos de entrenamiento como a la colección completa.

Tras este paso volveremos a entrenar el sistema de reconocimiento para continuar a partir de los nuevos datos de entrenamiento conseguidos, con esto el sistema debería tener más información, poder reconocer más imágenes y aumentar progresivamente la cantidad de datos que se tienen.

En todas las iteraciones se reconocerán todas las imágenes que originalmente no estuvieran etiquetadas. Esto lo haremos para que, si en iteraciones posteriores a reconocer una muestra se reconoce la misma con un mayor valor de confianza sustituiremos su primera aparición por esta de mayor confianza, únicamente haremos esto en el caso de mejorar la confianza de las muestras.

4.2.1. Cálculo de la confianza

En este apartado vamos a hablar de los dos métodos utilizados para determinar la confianza de las muestras y como serán los umbrales de confianza para cada uno de estos métodos.

El primero de estos métodos es el uso de la probabilidad a posteriori a partir del grafo de palabras de las muestras y el segundo es el que llamamos confianza por replicación o votación.

Confianza a partir del grafo de palabras

Para hablar de como calcular la confianza a partir de los grafos de palabras primero debemos dejar claro que es un grafo de palabras. Un grafo de palabras es un grafo acíclico dirigido con peso y etiquetas en sus aristas.

Esta estructura de datos representa un gran número finito de posibles secuencias de palabras de manera muy eficiente. El grafo de palabras es una versión podada del *trellis* de búsqueda de Viterbi obtenido en el proceso de reconocimiento.

Si extraemos el camino de mayor probabilidad del grafo nos quedamos con la secuencia que es la mejor hipótesis para una imagen que hemos reconocido, la probabilidad de esta secuencia es la que utilizaremos para determinar la confianza de la muestra.

La mayoría de la probabilidad suele acumularse en una o unas pocas posibles secuencias y en una o unas pocas posibles palabras, por lo que es razonable asumir valores elevados para establecer el umbral.

En este caso vamos a probar valores de 0.8 y 0.9, con el fin de asegurarnos de que el sistema tiene certeza de la decisión que toma sobre la secuencia.

Confianza por replicación o votación

El caso de la confianza por replicación o votación es sencillo, este consiste en que, en lugar de entrenar un único modelo o sistema, se entrenan varios, en nuestro caso 10. Si todos los modelos (o un número de ellos que determinemos) coinciden en la secuencia a reconocer para una muestra, esta es considerada como buena.

Los modelos utilizan una partición de los datos de entrenamiento para validación, un 10 % de las muestras, y el resto para entrenamiento. Lo que haremos en este caso es hacer 10 particiones y que cada modelo utilice una de ellas como validación y el resto como entrenamiento, con esto se utilizan todas las muestras repartidas entre los modelos.

Este método presenta una ventaja muy clara, que es que no necesita ningún método específico relacionado con como se ha realizado el reconocimiento, en este caso al darse una muestra como válida por votación o replicación de modelos es irrelevante como son estos modelos, sólo nos interesa la salida de los mismos.

Este método presenta dos desventajas respecto a otros métodos, la primera, es necesario entrenar un mayor número de modelos para cada iteración, en lugar de entrenar un modelo y obtener su confianza necesitamos entrenar x modelos para que simulen una votación sobre los posibles valores de una secuencia.

La segunda desventaja es que se pierde precisión al determinar la confianza, al usar un número finito y reducido de modelos es imposible estimar valores de confianza en intervalos específicos, una confianza de 8 o 9 modelos de acuerdo requiere entrenar al menos ese número de modelos, pero para simular un valor de, por ejemplo, 8.5 usando un decimal, necesitaríamos diez veces más modelos.

4.2.2. Introducción del carácter de error

Ahora vamos a añadir otro elementos a nuestro sistema de aprendizaje semi-supervisado, el carácter de error. El objetivo de esto es permitir aquellas líneas que superen el umbral total de confianza y que sólo un cierto porcentaje de sus símbolos no sean lo bastante fiables, aunque su mayoría sí lo sea, con el fin de no ser tan restrictivos.

Este carácter de error se introduce para sustituir a aquellos símbolos que no superen el umbral de confianza, con el fin de evitar utilizar información de la que no estamos seguros. En nuestro caso permitiremos que una secuencia tenga hasta un 20 % de símbolos de error para utilizarse en entrenamiento y considerarse para la colección completa.

La función principal de todo esto es que nos permite hacer el sistema más flexible, donde no se descarta una línea por uno o dos símbolos en los que se duda. Sin embargo esto introduce algunas líneas con símbolos incorrectos, en la sección de experimentos analizaremos en detalle como afecta introducir los mismos.

Lo que podemos decir aquí es que estos deberán ser corregidos manualmente antes de aceptar las muestras como buenas, aunque no es necesario corregirlas de cara al entrenamiento de los modelos.

Al introducir este símbolo el flujo de trabajo cambia ligeramente, aunque la idea general es la misma se introduce este último punto que nos hace necesario no tratar exactamente igual los datos con y sin errores, en la figura 4.4 podemos ver como es en este caso el flujo de trabajo.

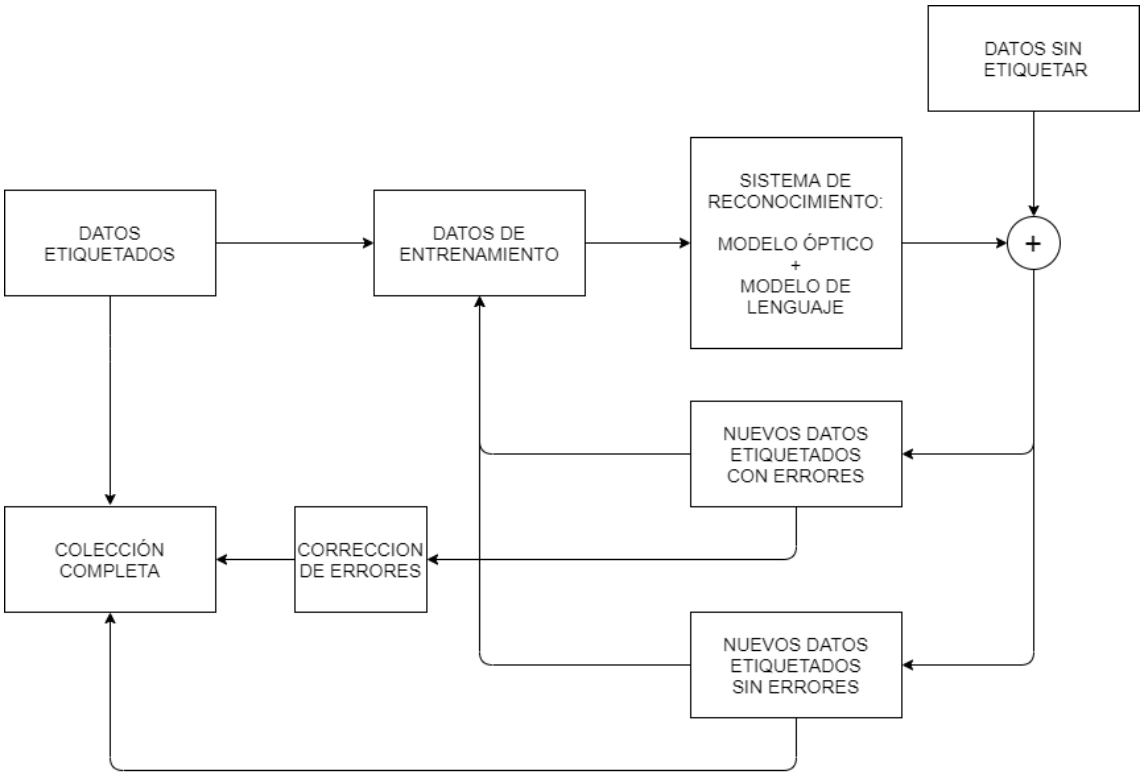

Figura 4.4: Esquema de funcionamiento para aprendizaje semi-supervisado con margen de error.

4.3 Interacción con el modelo

Vamos a pasar a hablar ahora de cuando es necesaria la interacción con el sistema, que es lo que el usuario debe hacer sobre las muestras y como determinar sobre que muestras debe actuarse.

Lo primero, en cuanto a cuando es necesaria la interacción, la utilidad que aporta la interacción es la de añadir información al sistema cuando este no tiene suficiente información para continuar con el etiquetado de muestras sin etiquetar.

En este caso vamos a plantear dos situaciones de interacción, un caso para las 4 primeras interacciones con el sistema y otro caso para el resto. Para las primeras 4 interacciones dejaremos que el sistema itere 20 veces y tras esto se hará la interacción. Para el resto de interacciones intervendremos cada 10 iteraciones del sistema semi-supervisado.

El método de interacción con el sistema es el siguiente, se determinan que 100 muestras son las que aportan menos información o son más difíciles de reconocer y (aunque en este caso lo haremos de manera automática) el operador humano etiqueta manualmente estas muestras. Hay que mencionar también que los bemoles se etiquetarán manualmente en todas las muestras, independientemente del nivel de confianza de las mismas y de la categoría a la que pertenezcan, así como se revisarán las claves manualmente debido a la importancia de las mismas.

Cabe destacar que para el caso en el que se permite cierto margen de error en el aprendizaje semi-supervisado también se deberán corregir aquellas muestras que contengan errores, aunque en estas únicamente hay que corregir estos símbolos de error, el resto de la muestra no es necesario revisarla.

Es una cuestión importante a tratar aquí ver cuál es el mejor método para determinar esas 100 muestras que deben ser etiquetadas, a continuación vamos a mostrar los dos métodos que planteamos en este trabajo.

4.3.1. Confianza como medida

Aquí utilizamos como medida para elegir estas 100 muestras a etiquetar la misma medida que utilizamos para determinar la confianza. La idea que subyace a esto es que, si una muestra tiene muy poca confianza, es porque el sistema carece de las capacidades para reconocer muestras de ese tipo, y su introducción al sistema deberá ser informativa.

Lo que debemos hacer es invertir el uso de las medidas de confianza, a continuación vamos a ver los dos casos de medidas de confianza que tenemos:

- **Confianza por probabilidad a posteiori**, aquí consideraremos los valores más pequeños de confianza a partir de la mejor hipótesis de cada una de las muestras que no han superado el umbral de confianza establecido.
- **Confianza por replicación o votación**, en este caso es algo más complicado, pues queremos buscar las muestras donde haya mayor conflicto entre los modelos entrenados. Para ello calcularemos la distancia de edición entre las diferentes muestras generadas por los modelos y etiquetaremos aquellas cuya distancia de edición acumulada sea mayor.

4.3.2. Entropía como medida

En este caso vamos a utilizar la entropía de las muestras como medida para determinar cuáles deben ser etiquetadas. La entropía es una medida de información, más concretamente una medida de la falta de información o incertidumbre.

El fin de esto es utilizar una medida más apropiada para la tarea que la confianza, ya que nuestro objetivo es extraer aquellas muestras que nos vayan a proporcionar más información, y esta medida nos lleva a las muestras para las que tenemos menos información, por lo que parece un método adecuado para determinar las muestras.

Para utilizar este método tenemos que calcular la entropía de cada una de las muestras que no están etiquetadas y no han superado el umbral de confianza y

quedarnos con las 100 muestras que tengan una mayor entropía, pues a mayor entropía mayor desinformación.

4.4 *Corpus* de datos

Vamos a ver ahora como es el conjunto de datos que tenemos. Lo primero, tenemos dos tipos de datos, los que están etiquetados y aquellos que no lo están, y los primeros se dividen en los que representan muestras reales y los que representan fragmentos.

El corpus es, de manera general, bastante limpio, entendiendo como esto que las imágenes no tienen ruido visual, y los elementos a observar en estas tienden a ser claros. En la figura 4.5 podemos ver como es una muestra cualquiera del *corpus*, en la figura 4.6 podemos ver un ejemplo de una de estas muestras fragmentadas, que no contienen información completa.

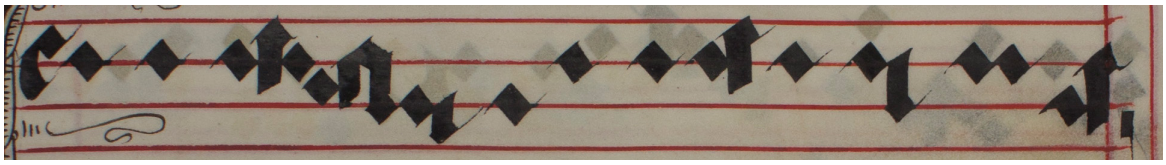

Figura 4.5: Muestra completa.

Figura 4.6: Muestra fragmentada.

Para el experimento base que demuestra que el sistema elegido es apto para la tarea utilizaremos únicamente los datos etiquetados originalmente. Estos datos también son los que se utilizarán para comenzar a entrenar los diferentes sistemas de aprendizaje semi-supervisado.

En la tabla a continuación podemos ver las características de este conjunto de datos base, con el número de muestras y símbolos en cada partición para el experimento base. Se utilizarán las tres particiones en conjunto al usarlos como datos iniciales de los sistemas, sólo se consideran separadas para el experimento base. La partición de entrenamiento contiene todos los símbolos que podemos encontrar en el conjunto de datos.

Partición	Nº líneas	Nº elementos	Nº símbolos
Entrenamiento	800	11569	19
Validación	200	2497	16
Test	97	1183	15

Tabla 4.4.1 Características del conjunto de datos base.

Vamos a extraer 350 muestras del conjunto de datos no etiquetados y las etiquetaremos manualmente, estas muestras son las que utilizaremos como conjunto de control, para asegurarnos de que el sistema mejora conforme se consiguen nuevos datos. En la siguiente tabla podemos ver las características de este conjunto.

Partición	Nº líneas	Nº elementos	Nº símbolos
Completa	350	8668	20

Tabla 4.4.2 Características del conjunto de datos de control.

Por último vamos a ver las características de los datos que originalmente están sin etiquetar, conocemos estos datos debido a que ya tenemos los mismos etiquetados, pero no podrían conocerse hasta haber completado el proceso de etiquetado. En la siguiente tabla podemos ver como son estos.

Partición	Nº líneas	Nº elementos	Nº símbolos
Completa	3002	74209	24

Tabla 4.4.3 Características del conjunto de datos sin etiquetar.

Un elemento interesante a analizar es como se distribuyen los datos en distintas dimensiones, por ejemplo, la cantidad de solapamiento de símbolos que tenemos entre nuestros 3 conjuntos de trabajo es importante, ya que si el solapamiento es muy bajo es posible que nos encontremos con algunos problemas de manera inicial por tratar de reconocer símbolos fuera de vocabulario. Esto podemos observarlo en la tabla a continuación.

Conjunto de datos	Nº símbolos únicos	Nº símbolos distintos de otros conjuntos
Base	19	0
Control	20	0
Sin etiquetar	24	4
Completo	24	4

Tabla 4.4.4 Solapamiento de símbolos entre los distintos conjuntos de datos.

Podemos observar como algunos símbolos del conjunto sin etiquetar se encuentran fuera de los datos con los que comenzamos a entrenar, por lo que estos no se podrán reconocer hasta que mediante la interacción con el sistema los introduzcamos al entrenamiento.

Lo último que queremos ver de los datos es si el tamaño de los mismos es uniforme, con tal de observar como de variables son los datos que tenemos. En la tabla a continuación se muestran las medias y desviaciones típicas para la longitud de las secuencias en cada uno de los conjuntos.

Conjunto de datos	Nº medio de símbolos por línea	Desviación típica
Base	13.9	7.9
Control	24.8	8
Sin etiquetar	24.7	8
Completo	22.05	9.3

Tabla 4.4.5 Tamaño medio de las líneas y desviación típica de cada conjunto de datos.

Podemos ver como, no sólo hay notables diferencias entre el número medio de símbolos por línea, sino que además nos encontramos con valores de desviación típica muy elevados. Esto no tiene por que ser un problema en nuestros experimentos pero es un elemento que tendremos en cuenta.

4.5 Herramientas

Vamos a utilizar distintas herramientas para desarrollar nuestros experimentos, al fin y al cabo este trabajo es un trabajo de investigación y no de desarrollo de una aplicación, por lo que aprovecharemos todo aquello que pueda sernos de utilidad.

La herramienta que vamos a utilizar para lo que respecta a nuestro modelo de lenguaje es PyLaia[1] [15]. Pylaia es una herramienta para aprendizaje profundo en análisis de documentos manuscritos basada en PyTorch.

Esta herramienta es un sucesor de Laia [2], introduciendo mejoras sobre la misma y escrita en un lenguaje de programación distinto, python, la cuál nos es familiar y por tanto nos facilita el trabajo con la herramienta.

Además, PyLaia permite un fácil trabajo con modelos de lenguaje, al incorporar diversos *scripts* que permiten una manipulación más sencilla de los datos de cara a combinar los resultados del modelo óptico con un modelo de lenguaje.

Con esta herramienta podemos crear un modelo óptico a partir de los parámetros deseados, entrenar dicho modelo a partir de nuestros datos de entrenamiento y por último extraer los resultados en un formato que podamos utilizar para, conjunto al modelo de lenguaje, obtener una transcripción para las imágenes.

Para la parte de la decodificación a partir de un modelo de lenguaje vamos a utilizar las herramientas Kaldi[3] y OpenFST[4].

Utilizaremos Kaldi para tratar prácticamente todos los elementos referentes a la creación, entrenamiento y aplicación de nuestro modelo de lenguaje, el cuál se forma mediante modelos ocultos de Markov.

[1] https://github.com/jpuigcerver/PyLaia
[2] https://github.com/jpuigcerver/Laia
[3] https://kaldi-asr.org/
[4] https://openfst.org/

La herramienta OpenFST está preparada para crea, combinar optimizar y buscar en en transductores de estados finitos ponderados o FSTs. Estos son autómatas con sus transiciones etiquetadas a la entrada y salida, así como ponderadas, muy utilizados en tareas del estilo de la que tratamos. Vamos a utilizar la herramienta en las últimas fases de trabajo sobre el modelo de lenguaje.

Por último comentar que utilizamos la herramienta utilizada en [19] para calcular la entropía de las muestras en el caso de utilizar esta como medida para determinar las muestras a etiquetar. Esta entropía se calcula a partir de los *lattices* de cada una de las muestras.

4.6 *Hardware*

Aunque se utilizan distintas piezas de hardware para la ejecución de estos experimentos hay un elemento clave en el contexto de las redes neuronales, este es la tarjeta gráfica. La importancia de esta viene de la potencia de cálculo de la misma, lo que permite tratar estos modelos de gran tamaño en un tiempo razonable.

El elemento del hardware no debería afectar a los resultados obtenidos, pues aunque en función de la potencia del mismo puede variar el tiempo de ejecución, al utilizar los mismos parámetros los resultados no deberían verse afectados.

En cualquier caso, es importante dar las especificaciones del hardware utilizado ya que al ser una pieza tan clave en la realización de los experimentos es como mínimo digna de mención.

Utilizamos dos tipos de tarjetas gráficas, el modelo GeForce RTX 1080 y el modelo GeForce RTX 2080, ambos producidos por el mismo fabricante y siendo de la misma serie, la serie 80, pero de distinta generación. Lo más importante en este caso es el tamaño de la memoria, ya que determina el tamaño de *batch* y esto puede afectar a algunos elementos del sistema, pero al tener ambos modelos 8GB de memoria podemos ignorar este elemento. En la figura 4.7 podemos ver con más detalle las características de estos modelos.

GPU	GEFORCE GTX 1080	GEFORCE RTX 2080
SMS	20	46
CUDA CORES	2.560	2.944
TENSOR CORES	n.a.	368
TENSOR FLOPS	n.a.	85
RT CORES	n.a.	46
TEXTURE UNITS	160	184
ROPS	64	64
RENDIMIENTO EN TRAZADO DE RAYOS	0,877 Gigarayos/segundo	8 Gigarayos/segundo
RENDIMIENTO RTX	8,9 billones de RTX-OPS	60 trillones de RTX-OPS
FRECUENCIA DE RELOJ DE LA GPU	1.733 MHz	1.800 MHz
FRECUENCIA DE RELOJ DE LA MEMORIA	5.005 MHz	7.000 MHz
MEMORIA DE VÍDEO TOTAL	8 GB	8 GB
INTERFAZ DE MEMORIA	256-bit	256-bit
ANCHO DE BANDA DE MEMORIA	320 GB/segundo	448 GB/segundo
TDP	180 W	225 W

Figura 4.7: Características del hardware utilizado.

CAPÍTULO 5
Experimentos

En este capítulo vamos a hablar de los experimentos realizados, veremos como evaluar los resultados de cada uno de ellos y trataremos de dar sentido a los mismos con el fin de, posteriormente, sacar unas conclusiones adecuadas. Además, incluiremos un apartado de experimentos sin éxito al final de este capítulo, donde hablaremos de otras técnicas y experimentos probados, complementarios al foco principal de nuestro trabajo.

5.1 Protocolo de evaluación

Lo primero que tenemos que ver es como vamos a evaluar los experimentos realizados. Al ser un trabajo que implica la colaboración de un sistema y un usuario es necesario medir tanto la calidad del sistema, por medio de medidas más tradicionales del error, como la cantidad de esfuerzo que debe hacer el usuario, con el fin de evaluar si nuestro sistema reduce el esfuerzo que debe hacer el mismo.

Por un lado, para evaluar la calidad de las transcripciones obtenidas, para hacer esto utilizamos una medida equivalente al Ratio de Error de Palabra (end inglés *Word Error Rate*, WER) pero utilizada en el campo del reconocimiento de música, esta es el al Ratio de Error de Símbolo (end inglés *Symbol Error Rate*, SER).

Generalmente se utiliza el SER y después se subdivide este en Ratio de Error de Altura (end inglés *Height Error Rate*, HER) y Ratio de Error de Glifo (end inglés *Glyph Error Rate*, GER) que consideran para cada nota la altura y el símbolo de la misma como dos componentes separadas. Sin embargo en los datos que nosotros tenemos sólo se considera la altura de los símbolos, pues la notación utilizada es para canto y no hay anotada una duración explícita asociada a las notas.

Por otro lado, para analizar el esfuerzo humano a la hora de evaluar la parte interactiva de nuestro sistema, esto lo vamos a medir de distintas maneras, con el fin de capturar toda la información posible.

Esta es una tarea ampliamente estudiada, y el esfuerzo humano se mide de distintas maneras dependiendo de la tarea que se realiza. En este caso nos vamos a limitar al número de veces que se interactúa, sin tener en cuenta el esfuerzo de cada interacción.

Nos limitamos a remitir al lector a una fuente donde se expande más en distintos métodos para medir estos resultados aunque aquí utilizamos únicamente los datos númericos. [20] A continuación vemos las distintas métricas utilizadas:

- Número total de líneas intervenidas: aquí calculamos cuantas líneas han necesitado de intervención, independientemente de si estas intervenciones eran de mayor o menor intensidad.

- Número medio de intervenciones por nueva línea anotada: aquí calculamos el número de intervenciones realizadas en función de la cantidad de líneas que se han anotado, utilizando esto como medida más general y con menor sesgo.

5.2 Proceso de entrenamiento y reconocimiento

Aquí vamos a describir en detalle como se realiza el proceso de entrenamiento y reconocimiento en una iteración normal de los sistemas de aprendizaje semi-supervisado, sin tener en cuenta las especificidades de cada uno.

La idea general está clara, los datos de entrenamiento de dicha iteración entrarán a los modelos para entrenar los mismos. Sin embargo, hay algún elemento más a considerar. Primero, para entrenar el modelo óptico particionaremos estos datos en entrenamiento y validación, un 80 % y un 20 %, y para no sobre-entrenar el modelo sobre los datos de entrenamiento, esta partición se hace aleatoriamente.

El modelo entrenará se iterativamente sobre los datos y se detendrá cuando el SER calculado sobre los datos de validación no mejore durante un número predefinido de iteraciones seguidas, en este caso ese número es 20.

Hemos definido para el entrenamiento del modelo un factor de aprendizaje de $3E$ - 4 e introduciremos distorsiones en los datos para ayudar al sistema a generalizar sobre el entrenamiento, además normalizaremos los datos a nivel de *batch*.

Tras esto se entrenará el modelo de lenguaje, en este caso sí, a partir de todos los datos que tenemos, pues éste no tiene el mismo riesgo de sobre-entrenamiento grave que tiene el modelo óptico.

Una vez tengamos ambos los utilizaremos en combinación para reconocer los datos de control y los datos sin etiquetar. Los datos de control después serán evaluados y se almacenará el SER obtenido, los datos que estaban sin etiquetar pasarán a la fase de cálculo de la confianza.

En el caso del cálculo de la confianza se tendrán que hacer las pertinentes operaciones en función del método a utilizar, y se reemplazarán los símbolos que no superen el umbral de confianza por el símbolo de error si utilizamos esta técnica. Determinaremos que muestras superan el umbral establecido y las añadiremos al conjunto de entrenamiento para repetir el proceso.

5.3 Experimento base

El primer experimento que debemos hacer es el experimento base, este es un experimento de control que consiste en repetir el experimento que encontramos en [6]. El objetivo de este experimento es asegurarnos de que los elementos de nuestro sistema de reconocimiento están bien calibrados de cara al resto de experimentos.

El sistema que vamos a utilizar para esto es sencillo, únicamente utilizaremos las partes del sistema de reconocimiento, el modelo óptico y el modelo de lenguaje, y entrenaremos a partir de las particiones del conjunto base para después reconocer el test de este mismo.

En la tabla 5.3.1 podemos ver los resultados de hacer una exploración de distintos valores para n y utilizando los dos posibles descuentos para el modelo de lenguaje de n-gramas.

Tamaño de n-grama	Descuento utilizado	Symbol Error Rate
2	KN	4.33
2	WB	4.21
3	KN	4.37
3	WB	4.29
4	KN	4.41
4	WB	4.42
5	KN	4.58
5	WB	4.50

Tabla 5.3.1 Resultados del experimento base.

Como podemos ver, en el mejor de nuestros resultados no sólo conseguimos estar a la altura del resultado esperado, sino que mejoramos este, lo que nos indica que el sistema planteado y los parámetros del mismo deberían ser adecuados para el resto de experimentos.

5.4 Experimentos de anotación del *Corpus* completo

En esta sección vamos a mostrar los diferentes experimentos realizados sobre la cuestión del aprendizaje semi-supervisado e interactivo. Veremos lo que hace distinto cada uno de estos experimentos, mostraremos en detalle los resultados obtenidos y analizaremos los mismos con el fin de determinar cuál es el mejor método de anotación.

5.4.1. Confianza por replicación o votación sin margen de error, selección de muestras mediante la confianza

Este experimento utiliza como medida de confianza la comparación de las muestras entre 10 modelos diferentes, utilizaremos esto para la parte de aprendizaje semi-supervisado, y para determinar las muestras a etiquetar en la parte interactiva utilizaremos la propia medida de confianza. Además, no aceptaremos muestras con símbolos que no superen el umbral.

Planteamos tres experimentos de etiquetado distintos, variando entre estos el umbral utilizado para la confianza:

En el primero necesitaremos que 9 de los 10 modelos estén de acuerdo en una muestra, en el segundo requeriremos que 8 de los 10 modelos estén de acuerdo en una muestra y para el último serán 9 de los 10 modelos teniendo que estar de acuerdo para las tres primeras iteraciones y 8 de los 10 teniendo que estar de acuerdo para el resto.

Cabe recordar también que el número de iteraciones de aprendizaje semi-supervisado comienza en 20 y se reduce a 10 a partir de la tercera iteración, coincidiendo con la reducción del umbral de confianza en el tercer experimento.

En las siguientes figuras 5.1 y 5.2 vamos a observar los resultados de estos experimentos, utilizamos el color azul para referenciar el experimento con umbral invariable a 9, el color verde para referenciar el experimento con umbral invariable a 8 y el color rojo para referenciar el experimento con umbral variable de 9 a 8.

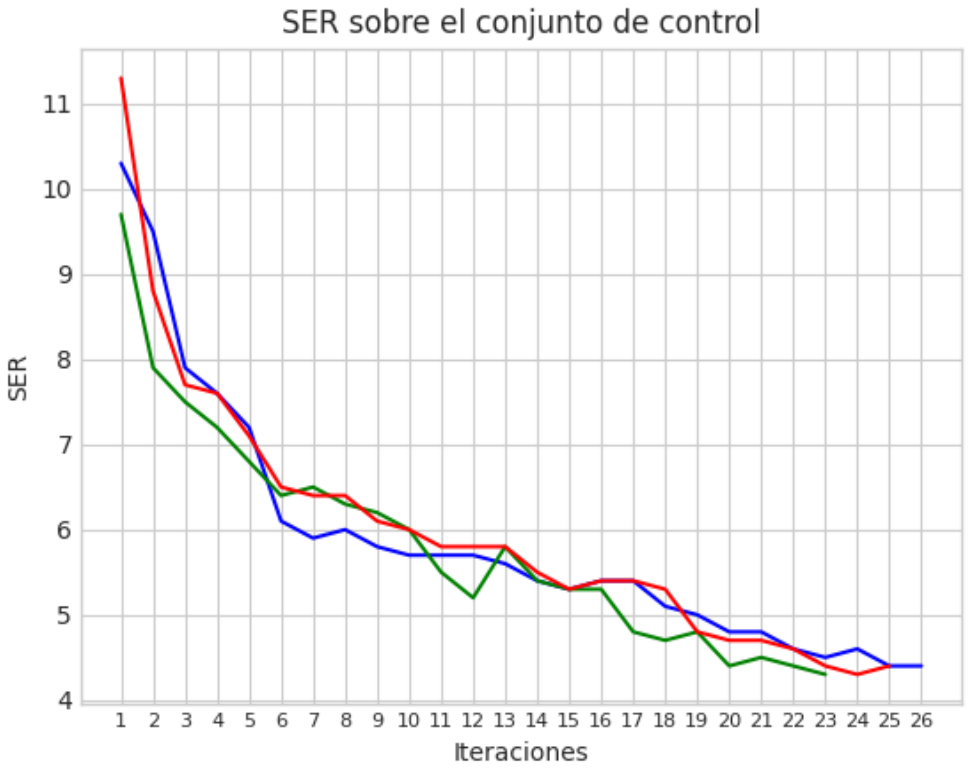

Figura 5.1: SER sobre el conjunto de control a lo largo del tiempo.

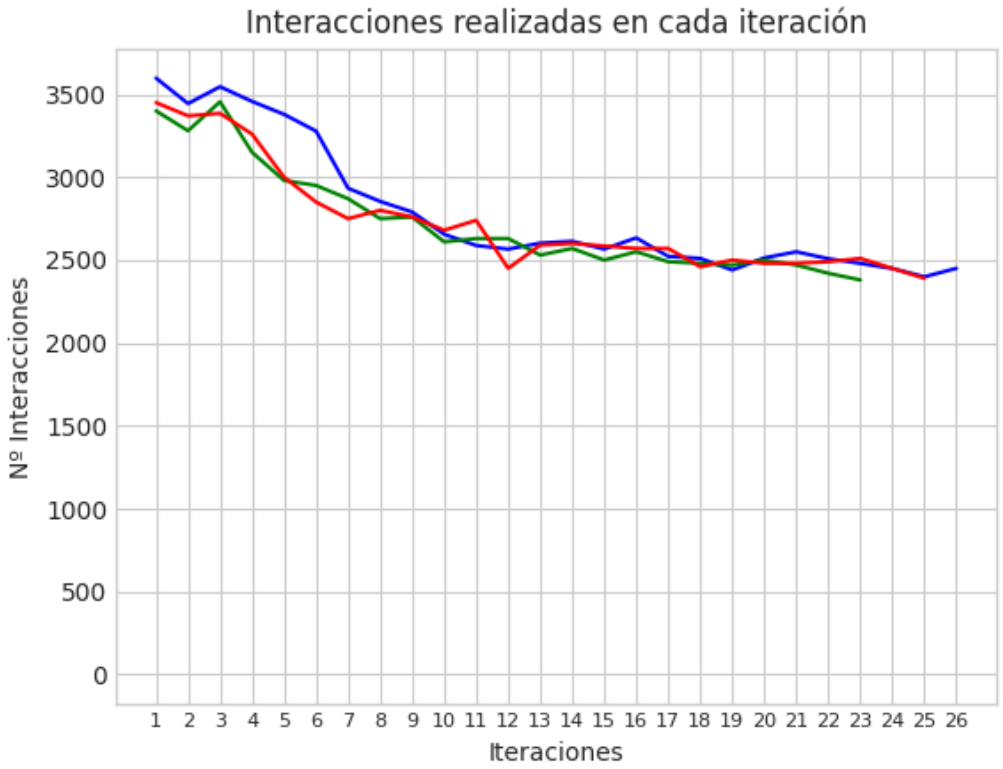

Figura 5.2: Número de interacciones realizadas por el operador humano en cada iteración.

Podemos ver como la diferencia más importante entre estos umbrales de confianza es como con umbrales más reducidos el número de iteraciones baja considerablemente. Esto sólo ocurre ligeramente en el caso de umbral variable, por lo que esto indicaría que las primeras iteraciones del sistema son muy importantes.

5.4.2. Confianza por probabilidad a posteriori sin margen de error, selección de muestras mediante la confianza

Este experimento utiliza como medida de confianza la probabilidad a posteriori calculada a partir de la mejor hipótesis del grafo de palabras, utilizaremos esto para la parte de aprendizaje semi-supervisado, y para determinar las muestras a etiquetar en la parte interactiva utilizaremos la propia medida de confianza. Además, no aceptaremos muestras con símbolos que no superen el umbral.

En este caso de nuevo se plantean tres experimentos distintos en función del umbral de confianza planteado:

Para el primero de ellos nos quedaremos con las muestras para las que tanto su confianza como la confianza de sus símbolos sea igual o superior a 0.9 o un 90 % de la probabilidad acumulada.

En el segundo experimento reduciremos el valor de este umbral a 0.8 o un 80 % de la probabilidad acumulada.

En el último caso de nuevo haremos un cambio en el umbral a partir de la tercera iteración, comenzando en 0.9 o 90 % y después reduciendo el mismo a 0.8 o un 80 %.

En las siguientes figuras 5.3 y 5.4 vamos a observar los resultados de estos experimentos, utilizamos el color azul para referenciar el experimento con umbral invariable a 9, el color verde para referenciar el experimento con umbral invariable a 8 y el color rojo para referenciar el experimento con umbral variable de 9 a 8.

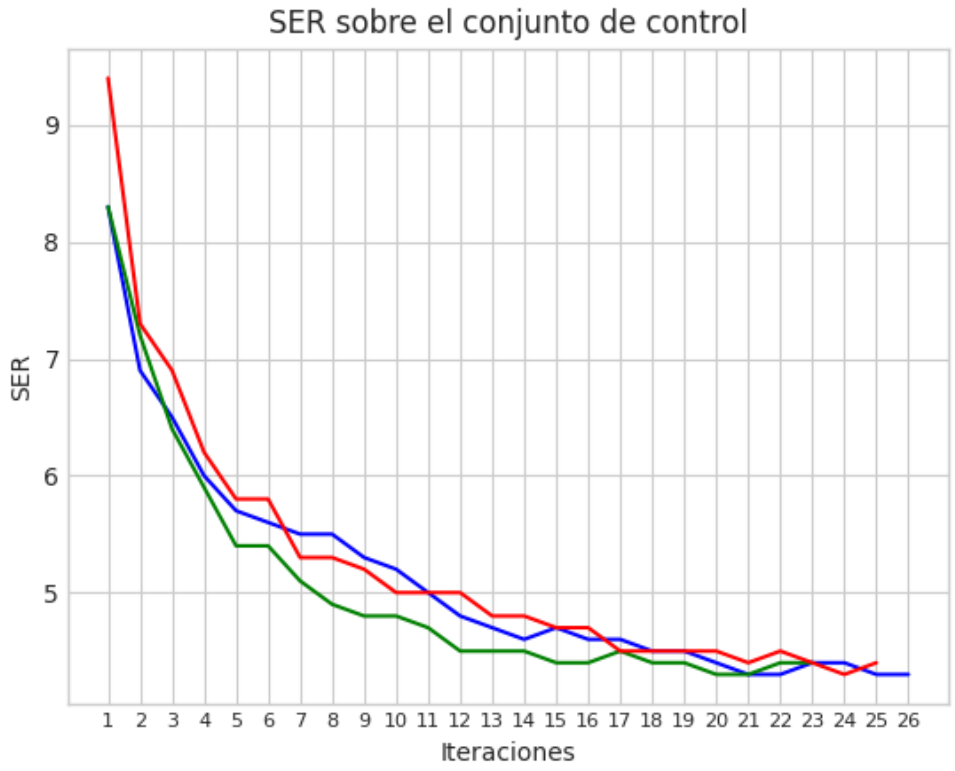

Figura 5.3: SER sobre el conjunto de control a lo largo del tiempo.

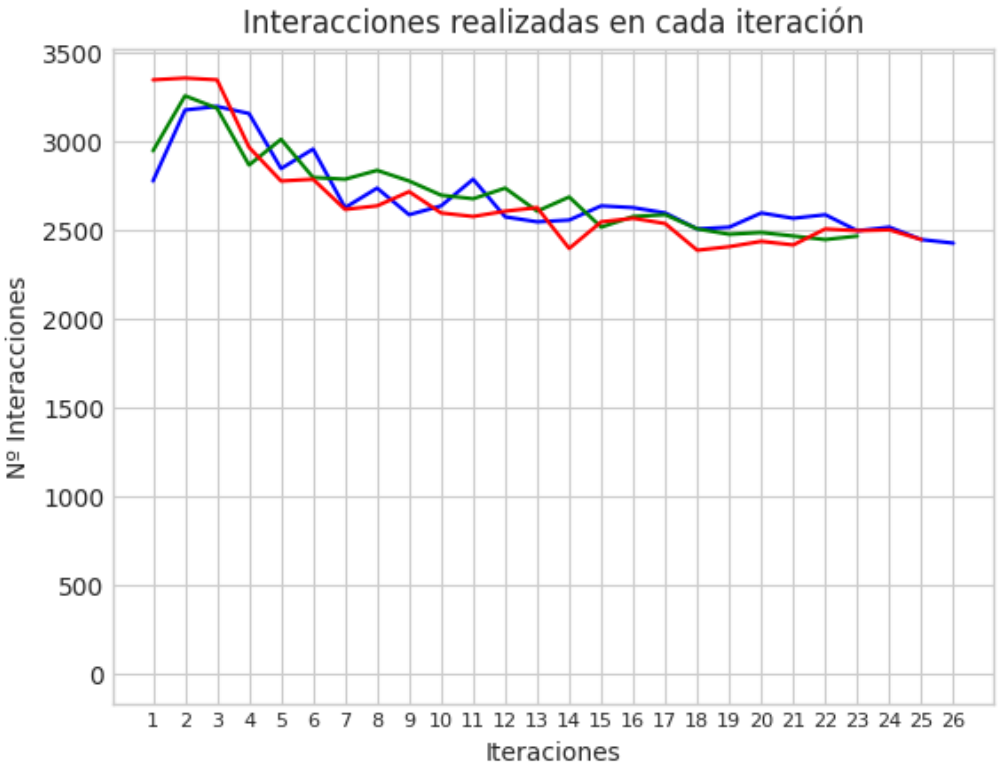

Figura 5.4: Número de interacciones realizadas por el operador humano en cada iteración.

Vemos como aquí se repiten unos resultados similares al primer experimento, de nuevo el factor a observar es como avanzan las iteraciones. Podemos ver también como el SER baja más rápidamente, llegando a bajar de 6 % entre 3 y 4 iteraciones antes.

Esto es un posible indicativo de que la calidad de las transcripciones obtenidas inicialmente es mejor, lo que haría esta medida de confianza más adecuada.

5.4.3. Confianza por probabilidad a posteriori con margen de error, selección de muestras mediante la confianza

Este experimento utiliza como medida de confianza la probabilidad a posteriori calculada a partir de la mejor hipótesis del grafo de palabras, utilizaremos esto para la parte de aprendizaje semi-supervisado, y para determinar las muestras a etiquetar en la parte interactiva utilizaremos la propia medida de confianza.

En este caso los símbolos que no superen el umbral serán sustituidos por el símbolo de error, aceptaremos las muestras que contengan menos de un 20 % de símbolos de error.

Se repite aquí el mismo patrón de los casos anteriores y realizaremos tres experimentos con tres umbrales de confianza distintos.

Mantendremos los mismos umbrales de confianza que en el experimento anterior, siendo 0.9, 0.8 y comenzando en 0.9 para después descenderlo a 0.8 en tres experimentos distintos.

Hay que tener en cuenta que estos umbrales se utilizan tanto para las muestras como para los símbolos, se podrían utilizar umbrales distintos para ambos casos, pero esto requeriría de muchas pruebas para establecer un valor adecuado y esto consumiría demasiado tiempo.

En las siguientes figuras 5.5 y 5.6 vamos a observar los resultados de estos experimentos, utilizamos el color azul para referenciar el experimento con umbral invariable a 9, el color verde para referenciar el experimento con umbral invariable a 8 y el color rojo para referenciar el experimento con umbral variable de 9 a 8.

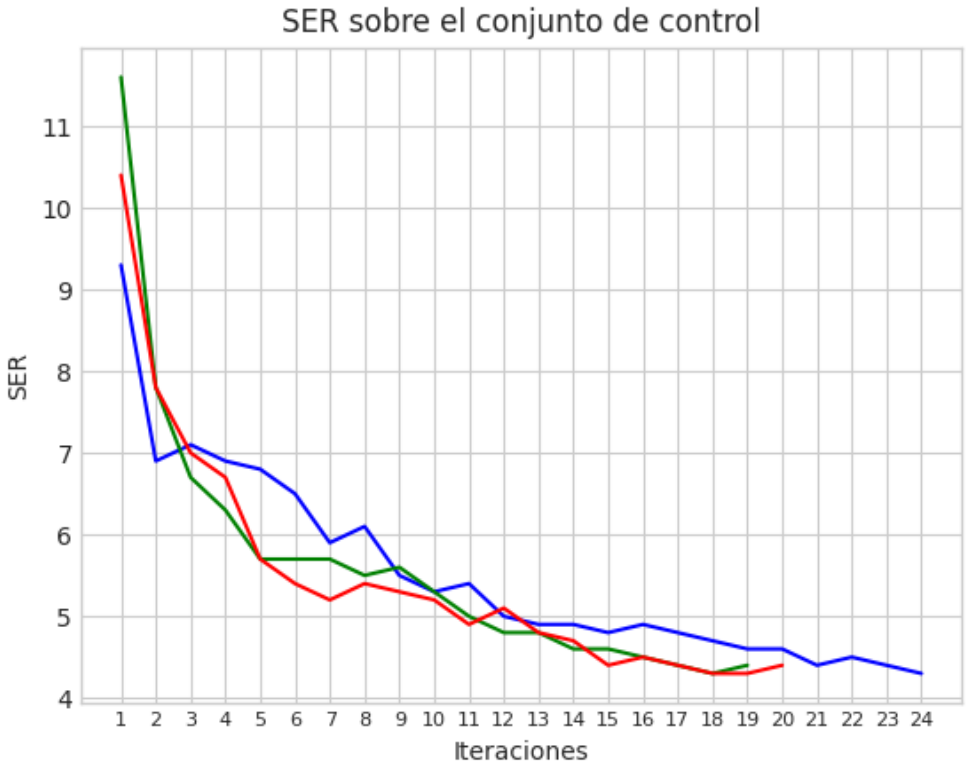

Figura 5.5: SER sobre el conjunto de control a lo largo del tiempo.

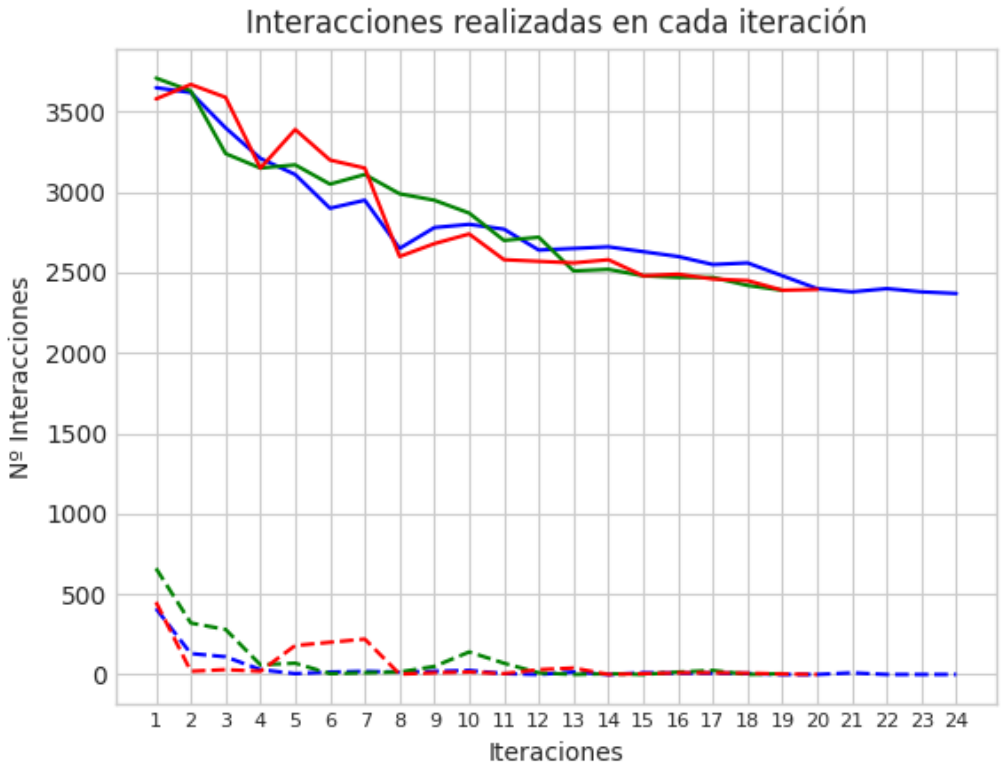

Figura 5.6: Número de interacciones realizadas por el operador humano en cada iteración.

Vemos como el número de iteraciones se reduce notablemente respecto a los experimentos anteriores. Podemos también observar como el error avanza de manera similar a los casos anteriores, lo que indica que la introducción del símbolo de error no afecta negativamente a esto.

Por último podemos destacar que la cantidad de interacciones que se realizan para solucionar estos errores no es notablemente grande. Esto indica que el esfuerzo humano no se ve aumentado lo suficiente como para que esto no merezca la pena.

Cuerpo del experimento	Umbral de confianza	Líneas anotadas manualmente	Líneas anotadas automáticamente	Interacciones / línea anotada
Replicación SE	0.9 -> 0.9	2588	414	21.3
	0.8 -> 0.8	2292	710	18.8
	0.9 -> 0.8	2495	507	20.5
A posteriori SE	0.9 -> 0.9	2591	411	21.2
	0.8 -> 0.8	2297	705	18.9
	0.9 -> 0.8	2489	513	20.4
A posteriori CE	0.9 -> 0.9	2399	603	20
	0.8 -> 0.8	1888	1114	15.9
	0.9 -> 0.8	1994	1008	16.8

Tabla 5.4.1 Resultados del los distintos experimentos, SE significa Sin Errores (sin margen de error), CE significa Con Errores (con margen de error).

Observamos un par de elementos interesantes si miramos las métricas mostradas en la tabla anterior.

El uso de la confianza por replicación o votación nos proporciona resultados casi idénticos al uso de la confianza a partir de la probabilidad a posteriori. Esto nos hace pensar que este método parece válido para marcar el umbral de confianza.

Aún en vista de esto el método muestra un problema que ya se veía claro al plantear el mismo, el tiempo de experimentación. Con este sistema se tarda 10 veces más que en el resto de experimentos, debido a que por cada iteración se entrenan 10 modelos.

Otra desventaja de este método es que el SER sobre el conjunto de control disminuye más lentamente, aunque finalmente se consigan alcanzar valores de SER similares al resto de los experimentos. Esto podría significar que la calidad de las transcripciones obtenidas a lo largo del proceso es peor.

Hay un patrón claro, la introducción del margen de error aumenta en gran medida la cantidad de líneas que se anotan automáticamente, y como podemos ver en los gráficos de barras, el número de interacciones adicionales es muy reducido en comparación con las interacciones a realizar por cada iteración.

Un elemento más a tener en cuenta es que anotar estos errores es mucho más sencillo de lo que pueda parecer a priori. Aunque para localizar un error en una secuencia parece necesario leer la misma de izquierda a derecha hasta llegar al error, al tener los *frames* donde se produce este error en el grafo de palabras podemos resaltarlos en la imagen para facilitar la tarea del operador humano. Esto se muestra en la figura 5.7.

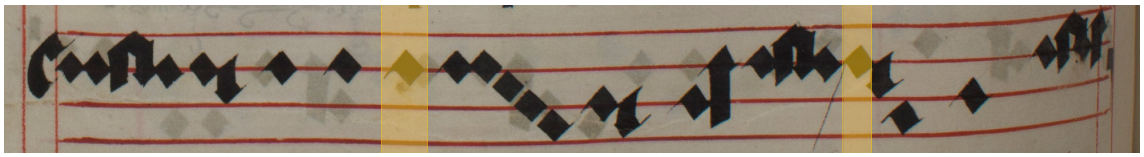

Figura 5.7: Imagen preparada para anotar por el operador humano, con los errores a anotar resaltados en amarillo.

En vista de los resultados obtenidos es claro como el uso de la confianza por probabilidad a posteriori con margen de error nos proporciona los mejores resultados. El error obtenido a partir de las muestras de control no es notablemente peor que en los otros dos casos y es el método en el que se obtiene una mejor eficiencia en cuanto a esfuerzo humano respecto al número de líneas anotadas.

Se podría plantear el caso en el que utilizamos la medida de confianza de votación o replicación de modelos y además se introduce el margen de error, pero, en vista del tiempo de cómputo requerido para este experimento y que no presenta resultados mejores no merece la pena realizar ese esfuerzo.

Ahora vamos a repetir el experimento de mejores resultados de los que acabamos de ver, cambiando el método de selección de muestras. Utilizaremos el uso del cálculo de la entropía en lugar de utilizar la misma medida de confianza para obtener estas, con el fin de observar si la entropía es una mejor medida de selección.

5.4.4. Confianza por probabilidad a posteriori con margen de error, selección de muestras mediante la entropía

Este experimento utiliza como medida de confianza la probabilidad a posteriori calculada a partir de la mejor hipótesis del grafo de palabras, utilizaremos esto para la parte de aprendizaje semi-supervisado, y para determinar las muestras a etiquetar en la parte interactiva utilizaremos la entropía de las muestras, quedándonos con aquellas muestras de mayor entropía. En este caso los símbolos que no superen el umbral serán sustituidos por el símbolo de error, aceptaremos las muestras que contengan menos de un 20 % de símbolos de error.

Como buscamos replicar un estudio previo utilizando un método distinto para seleccionar muestras en el apartado de aprendizaje interactivo realizaremos los mismos experimentos que se hicieron en ese. En este caso son tres experimentos considerando tres posibles umbrales, uno con umbral de confianza de 0.8, otro con 0.9 y un tercero comenzando en 0.9 y reduciéndolo a 0.8 a partir de la tercera iteración.

En las siguientes figuras 5.8 y 5.9 observaremos los resultados de estos experimentos, utilizamos el color azul para referenciar el experimento con umbral invariable a 9, el color verde para referenciar el experimento con umbral invariable a 8 y el color rojo para referenciar el experimento con umbral variable de 9 a 8.

Posteriormente compararemos estos resultados a observar con el mejor experimento anterior en la tabla 5.4.2, y analizaremos cuál es el mejor caso de anotado.

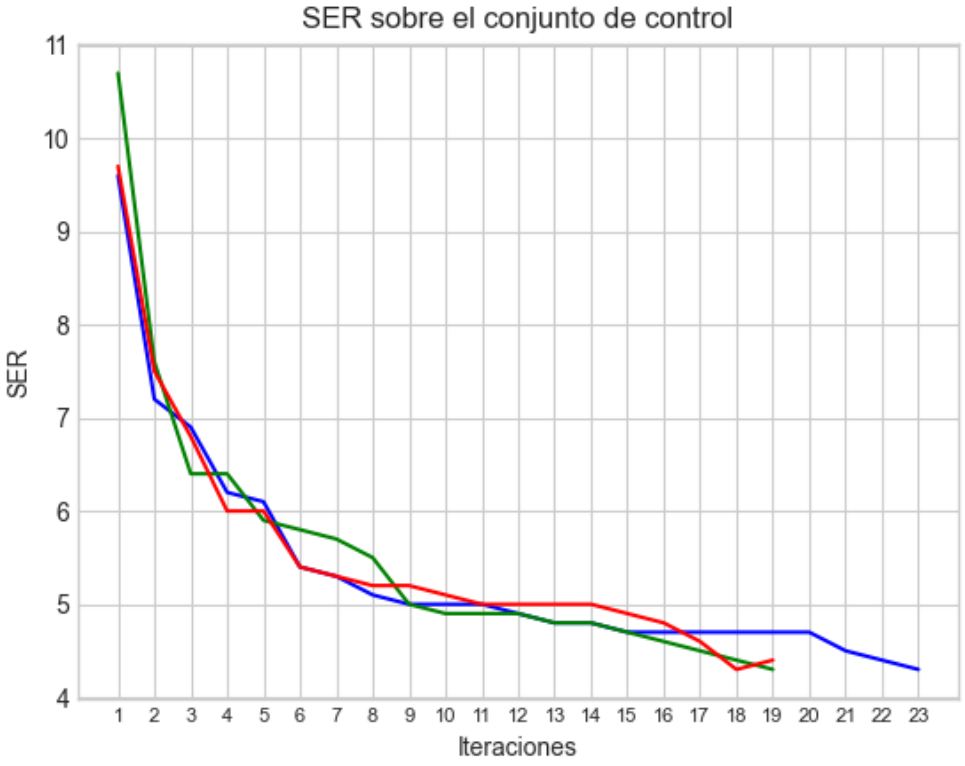

Figura 5.8: SER sobre el conjunto de control a lo largo del tiempo.

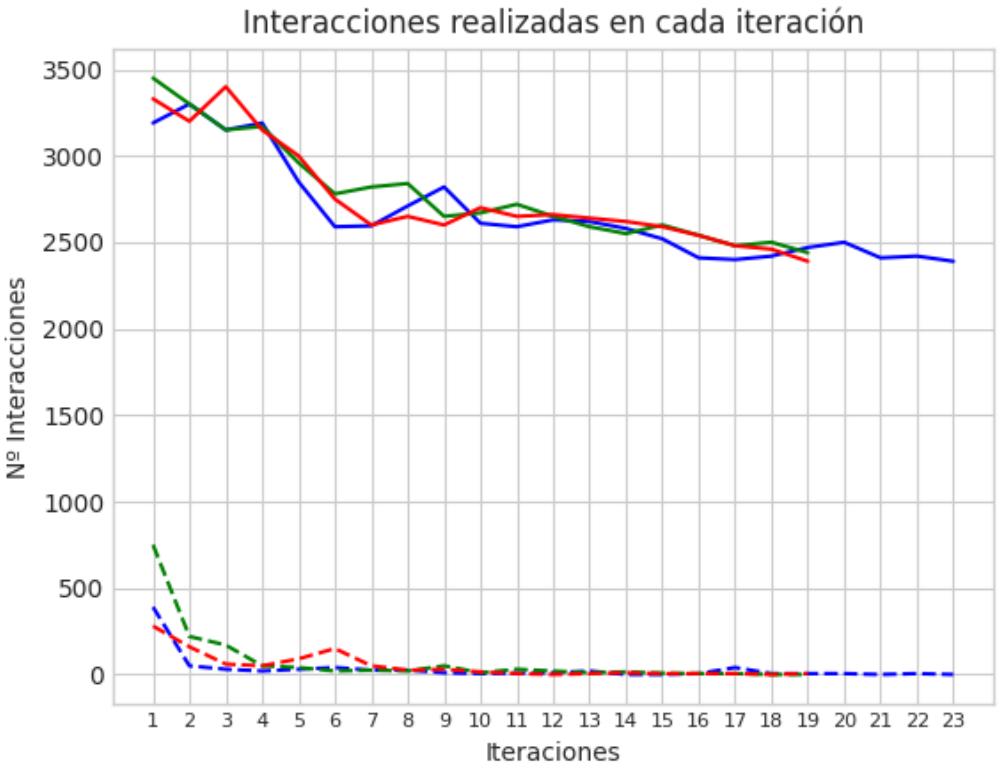

Figura 5.9: Número de interacciones realizadas por el operador humano en cada iteración.

Selección de muestras	Umbral de confianza	Líneas anotadas manualmente	Líneas anotadas automáticamente	Interacciones / línea anotada
Confianza	0.9 -> 0.9	2399	603	210
	0.8 -> 0.8	1888	1114	15.9
	0.9 -> 0.8	1994	1008	16.8
Entropía	0.9 -> 0.9	2287	715	19
	0.8 -> 0.8	1887	1115	15.5
	0.9 -> 0.8	1892	1110	15.7

Tabla 5.4.2 Comparación de los experimentos con confianza por probabilidad a posteriori con margen de error, con distintos métodos de selección de muestras.

Hay varios elementos de interés en la comparación, por un lado observamos como en 2 de los 3 experimentos se ha reducido el número de iteraciones necesarias para anotar el corpus.

Esto únicamente no ha sucedido en el experimento más laxo, donde es posible que al dejar suficiente margen de confianza no sea determinante el método de selección de muestras.

También podemos observar como en el caso en el que el número de iteraciones son iguales, que es el experimento donde el umbral de confianza es del 80 %, las interacciones a realizar por línea anotada disminuyen.

Esta disminución se debe a que utilizando este método se consigue un mayor número de líneas que no tienen ningún error o la cantidad de errores es menor, por lo que hay menos elementos que corregir.

El uso de la entropía como método de selección de muestras a anotar es prometedor, pues ya en este experimento ha mejorado los resultados para los tres casos planteados.

Esto no sólo nos permite dichas mejoras, sino que al ser una medida basada en la cantidad de información nos permitirá aplicar otros tipos de técnicas en el futuro basadas en la información de las muestras.

Hay que decir además de estos experimentos como el umbral de confianza más bajo es el que nos permite anotar los datos de manera más rápida, lo cuál no sorprende pues admitimos más muestras que en los otros casos.

Hay que mencionar que el SER del conjunto de control no muestra ser especialmente peor en este caso, por lo que parece razonable pensar que este umbral podría ser el mejor.

Con los experimentos que incluyen el margen de error se observa poca diferencia entre los experimentos con umbral de un 80 % y los experimentos que comienzan en un 90 % y después cambian a un 80 %.

Esto es porque muchas de las líneas que no superan el umbral del 90 % en las primeras iteraciones sí lo hacen al incluir el margen de error, y posteriormente si es necesaria esa disminución a un 80 % para mantener el ritmo de anotado.

No tenemos suficientes datos como para determinar que uno de estos umbrales sea estrictamente mejor que el otro, pero sí tenemos suficientes como para establecer que un umbral de un 90 % es demasiado restrictivo y disminuir el mismo no empeora notablemente los resultados.

También tenemos suficientes datos como para determinar que la introducción del margen de error mejora en gran medida los resultados a nivel de esfuerzo y anotación y mantiene los mismos a nivel de SER.

Por último decir que aunque con experimentación más extensa se podría ser mucho más concluyente respecto al método de selección de muestras, el uso de la entropía parece únicamente presentar ventajas respecto a la selección por confianza.

5.5 Experimentos sin éxito

En esta última sección vamos a mostrar dos líneas exploradas que no han tenido gran éxito, y que son secundarias a la investigación, por eso únicamente se mencionan aquí y no se han tratado de manera extendida a lo largo del trabajo.

Estas dos líneas son, por un lado tenemos que el símbolo que representa el bemol en las partituras es notablemente difícil de reconocer, por diversas razones que veremos más adelante, por lo que la primera línea se centra en torno a reconocer con mayor precisión este símbolo.

La segunda línea se centra en torno a tratar de combinar el reconocimiento de la letra de las canciones, que se encuentra bajo las partituras, con el reconocimiento de estas mismas partituras. El fin es ver si esto nos da una mejor calidad en las transcripciones de los datos al combinar dos fuentes de información.

5.5.1. Experimentos sobre bemoles

Para mejorar la precisión con la que se reconocen los bemoles vamos a introducir distintas modificaciones tanto al sistema de reconocimiento como al sistema de supervisión.

Primero hablaremos de las modificaciones al sistema de reconocimiento, entre estas tenemos duplicar las muestras que contienen bemoles en el entrenamiento y mantener un balance de muestras que contienen bemoles en entrenamiento y validación.

El primer experimento va a constar de aumentar la cantidad de muestras que contienen bemoles, hasta que estar representen un 20 % y un 30 % del entrenamiento.

Para esto calculamos cuantas muestras tenemos en entrenamiento y si la representación de muestras con bemoles no es la deseada se duplicarán estas en entrenamiento hasta tener el margen deseado.

Con esto buscamos hacer más incisión en el entrenamiento de estos símbolos, haciendo estos más apariciones en el entrenamiento.

Con el segundo experimento forzaremos a que haya un balance de imágenes que contengas bemoles en entrenamiento y validación.

El objetivo de esto es, ya que las muestras utilizadas en estos conjuntos se eligen aleatoriamente es posible que en una selección de muestras no haya ninguna con bemoles en la parte de entrenamiento, y el modelo resultante sea incapaz de reconocer los mismos.

El tercer experimento ha consistido en explorar las posibles transcripciones obtenidas y, a parte de priorizar estas en función de su medida de confianza, priorizaremos aquellas muestras que contengan bemoles.

Si encontramos una transcripción con bemoles que supera el umbral de confianza la preferiremos a otra transcripción que no los tenga aunque esa otra transcripción pudiera tener un valor más alto de confianza.

Esto introduce bemoles reconocidos erróneamente, por lo que utilizaremos técnicas externas para eliminar los mismos.

Sabemos bajo que condiciones aparecen los bemoles en este tipo de manuscritos, estos aparecen para "matar al diablo", esto es para suprimir un sonido desagradable en ciertas condiciones. Lo que haremos es eliminar los bemoles que aparezcan en una transcripción y no se den estas condiciones, donde ese bemol no está "matando al diablo".

Resultados

Aquí vamos a hablar de los resultados de estos tres experimentos:

Para el primero de nuestros experimentos los resultados han sido positivos en cuanto a la tarea que atacan, pues ha habido una mejora, aunque muy pequeña, en la cantidad de bemoles que se reconocen, sin embargo también han planteado un problema.

Lo que sucede es que al repetir una serie de muestras varias veces hay cierto sobre aprendizaje de las mismas y se observa como el sistema se ralentiza ligeramente, por lo finalmente no merece la pena esto.

Poniendo estos resultados en forma numérica observamos lo siguiente, durante las 4 primeras fases se anotan aproximadamente un 10 % menos de imágenes con suficiente confianza, debido a que al sobre-entrenar el sistema está menos seguro de las muestras que son muy distintas de sus datos de entrenamiento.

Esto se hace a cambio de un incremento en un 50 % del número de bemoles detectados, y aunque esto parece una mejora muy grande, en realidad ocurre porque originalmente eran muy pocos los bemoles etiquetados, empezando entorno a 10 y llegando en estos nuevos experimentos a unos 20.

Esto deja claro como a nivel de tiempo de etiquetado estamos empeorando el sistema, y la mejora en la cuestión de reconocimiento de bemoles no es suficiente, pues aunque hay un aumento en el número que reconocemos, este no es suficiente para que el intercambio merezca la pena.

Para el segundo de nuestros experimentos no se observa un cambio, los resultados no parecen verse modificados al asegurarnos de que los datos están balanceados entre entrenamiento y validación.

Esto no nos sorprende ya que, en teoría, el reparto aleatorio hecho originalmente es suficiente para que el reparto de muestras sea balanceado.

En vista de esto hemos mantenido esta medida a lo largo de los experimentos, debido a que no empeora los resultados y debería servir como medida de seguridad en el caso en que se diera la situación extrema de que el reparto de muestras sea el peor posible si fuera estrictamente aleatorio.

Para el tercer experimento nos encontramos con unos resultados interesantes:

Por un lado, el sistema de reconocimiento no empeora a simple vista, a pesar de que en algunos casos se eligen muestras de menor confianza.

Por otro lado la cantidad de bemoles que se reconocen aumenta, aunque de nuevo, no aumenta en una cantidad realmente determinante. La diferencia en *recall* entre utilizar esta técnica y no utilizarla es de un *recall* originalmente de 0.07 a un *recall* de 0.15- 0.2.

Vemos que el sistema de manera general no empeora y que conseguimos una mejora en el reconocimiento de bemoles, por lo que parece erróneo incluir este experimento en experimentos sin éxito, sin embargo, al atacar una parte del sistema de manera muy agresiva, sin hacer más pruebas no nos sentimos confiados como para decir que el mismo ha sido de éxito.

5.5.2. Experimentos combinando partituras y letra

De nuevo con el objetivo de mejorar las capacidades de reconocimiento de nuestros sistemas vamos a tratar de indagar en la tarea de utilizar nuestros dos flujos de información, música y letra, en lugar de utilizar sólo uno, la música.

Vamos a combinar las transcripciones de texto y música en una misma imagen, y trataremos de reconocer estas con el fin de observar la calidad de los resultados y determinar si esta línea de estudio es razonable.

Empezaremos por un subconjunto de los datos totales con el objetivo de reducir el tiempo requerido para realizar los experimentos.

Utilizaremos unos modelos óptico y de lenguaje con las mismas características que los utilizados en el resto de nuestros experimentos.

Este experimento no va a trabajar con el sistema completo de aprendizaje semi-supervisado, sino que se ejecutará únicamente la parte de reconocimiento.

Por otro lado, esta idea de utilizar varios flujos de información que se apoyan entre sí no es nueva, y está extendida por ejemplo en el reconocimiento de lenguaje de signos [1].

En estos casos se trabajaba con modelos de Markov, una tecnología que consigue en general peores resultados que las redes neuronales.

Aún con esto merece la pena probar si esta información añadida compensa el uso de una tecnología que requiere más trabajo para conseguir unos resultados parecidos.

Resultados

Los resultados en ambos casos han sido notablemente peores que en el caso del sistema que ya utilizamos, por lo que esta sección de resultados va a centrarse en analizar por qué esto ha sido así, más que en los resultados en sí.

En cuanto al primer experimento el SER, nuestra medida del error, se encuentra entorno al 80 %, esto es un valor muy alto, pues con el sistema normal nos encontramos entorno a un 5 % en lo que respecta a música y un 13 % en lo que respecta a texto, aunque este resultado no nos sorprende.

No nos sorprende porque hemos pasado de un vocabulario de 24 símbolos a 253 símbolos, donde muchos de ellos aparecen únicamente una o dos veces, por lo que los resultados son razonables.

A partir de aquí hemos aplicado una técnica similar a las utilizadas para música en [16, 23] con el fin de ver si esto mejora los resultados al simplificar el vocabulario y de la misma manera los datos, aunque sólo lo aplicaremos para símbolos que no tengan una presencia de al menos 5 apariciones.

Con esto se consiguen unos resultados de entorno a un 60 % de SER, que es una mejora notable respecto a los obtenidos anteriormente, pero todavía son mucho peores que reconocer los flujos de información por separado, por lo que esta técnica no es satisfactoria.

En lo que respecta a nuestro segundo experimento el principal problema es que con estos modelos ocultos de Markov conseguimos unos resultados claramente peores para los flujos separados.

Con esto el resultado es esperado, pues la unión de ambos tampoco consigue llegar a unos resultados que puedan ser de alguna manera competitivos con los obtenidos por las redes neuronales.

Aún con esto dicho, es una técnica interesante que podría tratar de adaptar o implementar con otras tecnologías con el fin de aprovechar dos flujos de información en para una misma transcripción.

CAPÍTULO 6
Conclusiones

6.1 Conclusiones respecto al procedimiento

Es claro que el procedimiento ha resultado exitoso, hemos conseguido anotar el conjunto de datos deseado y hemos podido ver como las técnicas introducidas consiguen mejorar le relación entre datos anotados y esfuerzo humano.

Comparando esto con experimentación más clásica, hemos seguido los mismos pasos que subyacen a esta como es debido. Hemos encontrado un problema, la falta de datos digitalizados para el estudio de música manuscrita, y hemos definido el mismo.

Se han estudiado los métodos habituales para resolver este tipo de problema, como es el uso de aprendizaje semi-supervisado e interactivo, y se han planteado hipótesis de como se podría mejorar esta tarea, planteando distintos métodos de cálculo de confianza y como afecta la introducción de un margen de error aceptable.

Hemos realizado una experimentación adecuada y bien limitada, donde buscamos poder explicar todo lo observado en la misma y respaldarlo con datos objetivos.

Por último, hemos realizado un análisis de estos resultados y hemos visto que métodos son los más adecuados y cuál es el comportamiento de las técnicas introducidas.

Nos hemos distanciado de las técnicas clásicas en el campo del reconocimiento de música manuscrita y hemos utilizado las tecnologías más novedosas para conseguir buenos resultados.

Sin embargo, no todo ha sido perfecto, el utilizar modelos pesados como son las redes profundas ha resultado en que los modelos en el caso del sistema de votación acaben consumiendo un elevado tiempo de computación.

Esto no hace menos válido este método, y el demostrar que puede encontrarse a la altura de otros métodos de evaluación en cuanto a calidad, aunque en lo que respecta a tiempo de cómputo sea peor, nos muestra que en otras tareas donde los modelos sean menos pesados o pudiendo optimizar el tiempo de cómputo este sería un buen método.

En general observamos que, en lo que respecta al procedimiento, este ha resultado satisfactorio y los resultados acompañan esta idea de manera positiva.

6.2 Conclusiones respecto a los resultados

Nuestros resultados muestran como anotando una media de 1890 imágenes, y partiendo de un conjunto ya anotado, conseguimos anotar un conjunto de imágenes sin etiquetar de 3002 imágenes. Esto supone que anotando únicamente un 63 % del corpus conseguimos que el mismo esté anotado al completo con transcripciones de gran calidad.

También mostramos como partiendo de un método estándar se consiguen anotar entorno a 510 imágenes de manera automática, anotando manualmente 2492, lo que supone un 83 % del conjunto de datos sin etiquetar.

Además se muestra como el uso de la replicación de modelos y votación también es una medida de confianza adecuada, pues aunque su tiempo de cómputo sea muy elevado, proporciona resultados de una calidad muy similar a los proporcionados a partir de la confianza por probabilidad a posteriori.

Con las distintas mejoras y técnicas introducidas se ha reducido este valor hasta el 63 % mencionado al principio, por lo que hemos conseguido una mejora de un 31.7 % en cuanto al esfuerzo humano necesario, sin empeorar la calidad de los resultados.

Podemos ver que para todos nuestros experimentos realizados el SER sobre el conjunto de control se acaba estabilizando entorno a un 4.3 - 4.4 de SER, muy similar al SER obtenido en nuestro experimento base, por lo que podemos entender que la calidad de los datos anotados está entorno a la misma de los datos originales.

Si observamos detenidamente las tablas de resultados del capítulo 5 podemos ver como la mejora en los resultados a nivel de esfuerzo humano realizado es gradual. Son distintos factores los que intervienen en estas mejoras en el resultado, siendo estos tales como la introducción del carácter de error y el determinar un umbral de confianza adecuado que mantenga la calidad de las transcripciones y reduzca el esfuerzo humano.

Ahora vamos a hablar en más detalle de la introducción del carácter de error, ya que esto ha sido lo que ha aportado las mejoras más considerables en lo que a reducir el esfuerzo humano se refiere.

6.2.1. Conclusiones sobre la introducción del carácter de error

Lo primero que tenemos que ver es como han afectado los cambios introducidos al sistema al añadir al mismo este carácter de error. Lo primero que observamos es que la cantidad de datos que se introducen al sistema aumenta considerablemente, pues hemos reducido las restricciones para aceptar una transcripción.

Esto también hace que en las distintas iteraciones en las que se exploran los datos mediante aprendizaje semi-supervisado, sin la intervención del operario

humano, el modelo aprenda el símbolo de error como algo a reconocer, ya que estos únicamente se eliminan cuando el usuario interactúa con el modelo.

Aquí nace una interesante pregunta, ¿es el hecho de que el sistema pueda producir símbolos considerador de error un problema? Aunque a simple vista pueda parecerlo en realidad que esto pueda ocurrir no introduce notables errores en el sistema.

En primer lugar el hecho de que el sistema interprete con mayor probabilidad que un símbolo en una imagen representa el error en lugar de una nota cualquiera quiere decir que el sistema tiene muchas dudas acerca de ese símbolo, por lo que etiquetarlo como un error sería adecuado.

En segundo lugar, estos son elementos que van a corregirse una vez llegue la fase de intervención por parte del operador humano y esto no ocurre con suficiente frecuencia como para que supongan un aumento notable del esfuerzo humano requerido.

A esto se añade que si no queremos que los distintos modelos puedan aprender a reconocer el símbolo de error será necesaria la interacción con el sistema cada vez que se vaya a entrenar un nuevo modelo, y aunque esto supone un número reducido de interacciones, también supone atención constante, cosa que preferimos evitar.

Por último hay que destacar que corregir estos errores es muy sencillo, como ya hemos mostrado en la figura 5.7, porque podemos resaltar los elementos a corregir a partir del grafo de palabras, reduciendo el esfuerzo que podría suponer la búsqueda de símbolos individuales en una secuencia.

6.3 Cumplimiento de objetivos

En esta sección vamos a pasar sobre cada uno de los objetivos planteados al comienzo del trabajo y estableceremos si hemos conseguido cumplir los mismos y/o bajo que condiciones se ha hecho esto.

El primero de los objetivos planteados es "Desarrollo e implementación de un sistema de aprendizaje semi-supervisado". Para esto como los distintos subobjetivos indicaban hemos utilizado los sistemas de reconocimiento de música manuscrita más novedosos y actuales, hemos sido capaces de calcular los distintos métodos de confianza, no sólo los basados en grafos de palabras y hemos podido automatizar el proceso para facilitar los experimentos.

Con todo esto visto y siendo el caso en el que con este sistema hemos podido anotar el conjunto de datos al completo está bastante claro como estos objetivos los hemos cumplido con creces.

El segundo objetivo planteado es "Desarrollo y uso de un sistema de aprendizaje interactivo". Aquí hemos estudiado varios métodos para determinar cuando se debe actuar, utilizando distintas maneras de calcular la confianza así como la entropía para decidir las muestras a etiquetar, además hemos planteado como debe utilizarse la idea de un carácter de error con el fin de hacer más fácil el trabajo del operador humano.

Podemos ver con claridad como los objetivos planteados también se han superado en este caso, así mismo, también hemos creado distintos *scripts* que nos permiten automatizar la interacción para simular el trabajo humano, facilitando la ejecución del sistema. Aunque esto no formara parte de los objetivos ha sido importante.

En tercer lugar tenemos una serie de objetivos que nos permiten obtener resultados y evaluar el trabajo realizado. Los dos primeros de estos se refieren a experimentación, y consideramos haber logrado ambos al conseguir, por un lado, unos resultados mejores de los esperados en el experimento base y por otro lado al conseguir anotar el conjunto de datos con todos y cada uno de los métodos utilizados.

Por último, consideramos el análisis de resultados adecuado pues hemos mostrado con datos objetivos que sistemas han ido mejor y hemos buscado dar explicación a estos resultados. Aunque es difícil determinar numéricamente si todos los objetivos se han cumplido, en vista de lo mostrado, consideramos que hemos podido lograr lo propuesto.

6.4 Relación con los estudios cursados

Es fácil ver como se relaciona este trabajo con los estudios cursados en el máster con simplemente observar cada una de sus partes. Por un lado tenemos componentes afines a la asignatura de Reconocimiento de Escritura, pues estamos utilizando un sistema de reconocimiento de música manuscrita, que aunque no sea texto sino música los métodos utilizados tienen el mismo fundamento.

Asimismo en esta asignatura se ven y utilizan los mismos grafos de palabras que hemos utilizado para estimar la confianza a partir de la probabilidad a posteriori, por lo que la relación con esta asignatura es clara.

También hemos estudiado el aprendizaje activo e interactivo en el máster, concretamente en la asignatura de Predicción Estructurada Estadística, por lo que ahí encontramos otra relación con los estudios.

Además de esto, el uso de técnicas como es el cálculo de la entropía derivacional es un elemento que se enseña en la asignatura de Aprendizaje Automático Avanzado, por lo que también tenemos relación con la misma en nuestro trabajo.

Con todo esto tenemos además que el trabajo en sí utiliza técnicas de inteligencia artificial y reconocimiento de formas que se enseñan en distintas asignaturas del máster como son las redes neuronales, las redes recurrentes y demás elementos, por lo que es claro que el trabajo está fuertemente relacionado con los estudios cursados.

CAPÍTULO 7
Trabajos futuros

A continuación vamos a hablar de las distintas líneas de trabajo por las que podría continuar un trabajo como este, dando algo de detalle sobre las mismas y el por qué son de interés.

7.1 Número de muestras adicionales no estático

A lo largo del desarrollo del trabajo una constante ha sido el número de muestras que se etiquetaban en la fase de aprendizaje interactivo, en todas las iteraciones se han etiquetado 100 nuevas imágenes.

Como se puede ver en los resultados, en repetidas ocasiones 100 imágenes no contienen suficiente nueva información como para que el sistema etiquete un gran número de imágenes adicionales de manera automática.

Esto nos indica que la decisión de etiquetar un número fijo de imágenes no es adecuado, por lo que sería mucho más interesante utilizar algún tipo de medida de información con el fin de seleccionar un número de imágenes que nos aporten esa cantidad de información.

Es decir, en lugar de fijar un número de imágenes medir cuanta información puede aportar cada una de las imágenes sin etiquetar y elegir imágenes hasta que lleguemos a un umbral de información total a partir del cuál etiquetaremos y continuaremos con el proceso.

La pregunta en este caso sería, como medimos la información, que hacemos para determinar cuanta información aporta cada muestra y como determinamos cuál es el umbral de información necesario para que el sistema funcione de manera adecuada.

Una opción es continuar utilizando la entropía como medida, dado que es una medida de desinformación sabemos que aquellas muestras menos informadas serán las que deberían apoyar más eficientemente al sistema de reconocimiento.

En este caso determinaríamos un umbral de entropía acumulada, la mejor manera de determinar un umbral adecuado sería la prueba y error, probar distintos umbrales con conjuntos de datos de prueba para evaluar cuál de estos es razonable.

Con este umbral determinado etiquetaríamos muestras y acumularíamos la entropía de estas hasta llegar al umbral en cada iteración, entonces añadiríamos estas muestras al conjunto de entrenamiento y continuaríamos con el proceso.

7.2 Generación de datos artificiales

Uno de los problemas con los que nos hemos encontrado en este trabajo es con símbolos que aparecen pocas veces y cuyas apariciones son bastante diversas unas de otras, como es el caso de los bemoles, por lo que una línea de trabajo sería atacar esto.

En este trabajo generamos distorsiones y modificamos los datos para obtener otras muestras y hacer el sistema más robusto, esto es un método de generación de datos conocido y probado en múltiples campos y que ha dado buenos resultados tanto en esos campos como en este.

Sin embargo, existe una opción que no hemos puesto a prueba, esto es, la generación de datos completamente artificiales que nos ayuden a tratar el problema con el que nos encontramos. Por ejemplo para el caso de los bemoles, generar muestras artificiales que contengan los mismos, con el fin de que el sistema aprenda mejor a reconocerlos.

Este trabajo puede ser tanto manual como automático, pues se pueden crear manualmente muestras que contengan el símbolo con el que tenemos problemas para añadirlas al sistema, aunque este método es menos preferible ya que requeriría de una gran cantidad de esfuerzo humano.

La idea de generar datos artificiales de manera automática es una idea mucho más interesante, ya que nos permitiría obtener modelos ópticos de mucha mayor potencia y calidad, sin embargo, esta no es una tarea sencilla. Hay un gran número de elementos a tener en cuenta, como por ejemplo la variabilidad de la representación de los datos, al ser manuscritos, que debe estar también presente en estos datos artificiales. Además, queremos datos que estén en el mismo estilo de escritura que los datos que vamos a reconocer.

La generación de datos de manera artificial se ha puesto a prueba previamente, sin embargo, esto se ha utilizado en estudios estadísticos para enriquecer los datos, no hay tantas instancias de su uso en reconocimiento de texto o en reconocimiento de música. Y bajo estas condiciones es difícil de extrapolarlo a nuestra tarea.

En general, es una tarea muy interesante, pues un sistema capaz de generar datos artificiales similares a los originales y que nos permita perfeccionar los modelos ópticos sería un gran avance no sólo en música sino en general en el campo del reconocimiento de imágenes, pero también es una tarea difícil, por lo que requeriría mucho trabajo.

7.3 Combinar letra y música

En este trabajo hemos explorado algunos métodos para utilizar dos flujos de información con la intención de que estos se apoyen entre si, y aunque no hemos conseguido grandes resultados la idea sigue siendo interesante y merece una investigación propia.

Además, por como son los símbolos resultantes en nuestro primer experimento al respecto, se puede observar como nuestra idea de que hay algún tipo de relación entre las notas musicales y las letras no es infundada.

La idea más prometedora que hemos visto en este trabajo es la planteada con los modelos ocultos de Markov, donde se utilizaba de manera explícita la información de una de las imágenes para apoyar la otra. Un posible camino sería explorar esta misma idea para redes neuronales o modelos de aprendizaje profundo.

Esta tarea es complicada, pues plantea dos problemas, el primero, plantear como debe realizarse la conexión de dos redes, tanto en entrenamiento como inferencia, y como debe tratarse la retro propagación del error en un sistema como este es complicado. El segundo problema que plantea es, una vez tenemos resuelta la parte teórica hay que hacer la práctica, este trabajo es de implementación y creación de una tecnología, y esto claramente presenta su propio desafío.

Con esto conseguiríamos poder aplicar la técnica utilizada en otras tecnologías que nos permite combinar flujos de información y aplicarla sobre las tecnologías más punteras que están al alcance de nuestra mano hoy en día.

7.4 Indexación y búsqueda

Otro paso a realizar sobre el trabajo hecho aquí es crear y utilizar un indexador para poder indexar el conjunto de datos de manera que se pueda hacer búsqueda sobre el mismo.

Sin embargo, la indexación de música presenta sus propios desafíos, por ejemplo si la comparamos con la indexación de texto, cuando tratamos esta tenemos en cuenta dos elementos, carácteres y palabras, que nos permiten hacer búsquedas más o menos específicas en función de nuestras necesidades.

En el caso de la música sólo podemos hacer las búsquedas a nivel de símbolo, y si queremos buscar una secuencia en un conjunto de datos de gran tamaño el coste computacional de las distintas combinaciones de símbolos hasta encontrar la secuencia indicada es muy elevado.

Es por esto que es de interés encontrar patrones melódicos, utilizar tanto conocimientos en música como técnicas que nos permitan aprender las relaciones entre los elementos de un conjunto de datos para aprender como se pueden o deben agrupar los elementos musicales con el fin de facilitar la búsqueda de los mismos.

Con esto se podría construir un sistema capaz de realizar búsquedas en una base de datos de manera eficiente, lo que haría maś accesible este tipo de datos para el público general.

Bibliografía

[1] M. Brand, N. Oliver, and A. Pentland. Coupled Hidden Markov Models for complex action recognition. *CVPR*, 1997.

[2] J. Calvo-Zaragoza and I. Barbancho and L.J. Tardón and Ana M. Barbancho. Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications, Volume 18*, 933–943 2015.

[3] J. Calvo-Zaragoza and A. H. Toselli and E. Vidal. Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks. *Pattern Recognition Letters, Volume 128*, 115-121 2019.

[4] J. Calvo-Zaragoza and A. H. Toselli and E. Vidal. Early Handwritten Music Recognition with Hidden Markov Models. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 319-324 2016.

[5] J. Calvo-Zaragoza and A. H. Toselli and E. Vidal. Handwritten Music Recognition for Mensural Notation: Formulation, Data and Baseline Results. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 1081-1086 2017.

[6] J. Calvo-Zaragoza, A. H. Toselli, E. Vidal and J. A. Sánchez. Music Symbol Sequence Indexing in Medieval Plainchant Manuscripts. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 882-887 2019.

[7] Castelli, V. & Cover, T. M. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Trans. on Information Theory*, 42, 2102–2117 1996.

[8] Fabio Gagliardi Cozman, Ira Cohen and Marcelo Cesar Cirelo. Semi-Supervised Learning of Mixture Models. *In Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03). AAAI Press*, 99–106 2003.

[9] Graves A. Long Short-Term Memory. In: Supervised Sequence Labelling with Recurrent Neural Networks. *Studies in Computational Intelligence, Volume 385*. Springer, Berlin, Heidelberg, 2012.

[10] Graves, A. and Fernández, S. & Gomez, F. & Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *Proceedings of the 23rd International Conference on Machine Learning*, 369–376 2006.

[11] Hasan, A. S. M and Islam, Saria and Rahman, M. A Comparative Study of Witten Bell and Kneser-Ney Smoothing Methods for Statistical Machine Translation. *JU Journal of Information Technology (JIT), Volume 1* 2012.

[12] R. Hecht-Nielsen. Theory of the Backpropagation Neural Network. *Neural Networks for Perception*, 65-93 1992.

[13] Hochreiter, Sepp and Schmidhuber, Jürgen. Long Short-Term Memory. *Neural Computation, Volume 9*, 1735-1780 1997.

[14] Kingma, Diederik P., Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *In Advances in neural information processing systems*, 3581-3589. 2014.

[15] Mocholi C. Development and experimentation of a deep learning system for convolutional and recurrent neural networks. Universitat Politècnica de València. 2018.

[16] Nuñez-Alcover, A. & de León, P. J. & Calvo-Zaragoza, J. Glyph and Position Classification of Music Symbols in Early Music Manuscripts. *Pattern Recognition and Image Analysis, LNCS, Volume 11868* 2019.

[17] Rebelo, A. and Fujinaga, I. and Paszkiewicz, F. and Marçal, A. and Guedes, C. and Cardoso, J. Optical music recognition: State-of-the-art and open issues. *International Journal of Multimedia Information Retrieval, Volume 1*, 173–190 2012.

[18] Reinhard Kneser and Hermann Ney. Improved backing-off for M-gram language modeling. *1995 International Conference on Acoustics, Speech, and Signal Processing, Volume 1*, 181-184 1995.

[19] V. Romero, J. A. Sánchez and A. H. Toselli. Active Learning in Handwritten Text Recognition using the Derivational Entropy. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 291-296 2018.

[20] Romero, Verónica, Alejandro Héctor Toselli, and Enrique Vidal. Multimodal interactive handwritten text transcription. *Vol. 80. World Scientific*, 2012.

[21] Stanley F. Chena and Joshua. Goodman An empirical study of smoothing techniques for language modeling. *Computer Speech & Language, Volume 13*, 359-394 1999.

[22] Timothy C. Bell, John G. Cleary, Ian H. Written Text compression. 1990.

[23] M. Villarreal and J. A. Sánchez. Handwritten Music Recognition Improvement through Language Model Re-interpretation for Mensural Notation. *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 199-204 2020.

[24] Williams, R. J and Zipser, D. Gradient-based learning algorithms for recurrent networks and their computational complexity. *Backpropagation: Theory, architectures, and applications, Chapter 13* 1995.

[25] B. Xu and N. Wang and T. Chen and M. Li. Empirical Evaluation of Rectified Activations in Convolutional Network. 2015.

[26] Zhu, Xiaojin & Lafferty, John & Ghahramani, Zoubin. Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. 2003