UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Reconocimiento automático de datos numéricos metereológicos en documentos históricos impresos representados en forma de tablas

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: David Villanova Aparisi

Cotutores: Joan Andreu Sánchez Peiró
José Miguel Benedí Ruiz

Curso 2019-2020

Resum

En el nostre treball es presenta un model de transcripció automàtica de taules escrites a màquina amb informació de tipus numèric. Aquest model es basa en l'ús de xarxes neuronals artificials multicapa i ferramentes de processament d'imatge, així com en la utilització de múltiples tècniques que augmenten la precisió del sistema. A fi d'obtenir millors resultats, s'ha triat l'elaboració de múltiples models de llenguatge que aporten informació contextual durant la transcripció. En el present document es discuteix la metodologia seguida així com els diferents models sobre els quals s'ha iterat, comparant els resultats obtinguts.

Paraules clau: reconeixement de text, offline, dades numèriques, model de llenguatge

Resumen

En nuestro trabajo se presenta un modelo para la transcripción automática de tablas escritas a máquina con información de tipo numérico. Dicho modelo se basa en el uso de redes neuronales artificiales multicapa y herramientas de procesamiento de imagen, así como en el empleo de múltiples técnicas que aumentan la precisión del sistema. A fin de obtener mejores resultados, se ha probado la elaboración de múltiples modelos de lenguaje que aporten información contextual durante la transcripción. En el presente documento se discute la metodología seguida así como los distintos modelos sobre los que se ha iterado, comparando los resultados obtenidos.

Palabras clave: reconocimiento de texto, offline, datos numéricos, modelo de lenguaje

Abstract

In our work we present a model for the automatic transcription of typed tables with numerical data. Such model is based on the usage of multilayer artificial neural networks and image processing tools, as well as the handling of multiple techniques which increase the precision of the system. For the purpose of obtaining superior results, the construction of multiple language models, which provide contextual information during the transcription, has been tried. In the following document we discuss the followed methodology as well as the different models that have been taken into account, analysing the obtained results.

Key words: text recognition, offline, numerical data, language model

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

El reconocimiento automático de textos escaneados, a lo cual nos referiremos como transcripción automática, persigue la obtención de un documento digital a partir del texto contenido en una imagen.

1.1 Motivación

Los motivos para centrarse en la transcripción automática, en nuestro caso de textos escritos a máquina en forma de tablas, son varios. Por una parte, hay una ingente cantidad de información no digitalizada que es vulnerable al paso del tiempo y menos accesible. La transcripción automática sirve como herramienta para garantizar la accesibilidad y preservación de la información histórica.

Por otra parte, los datos[1] que tratamos pueden habilitar un análisis posterior que revele más información sobre el impacto del cambio climático. Old Weather [1], iniciativa que nos ha cedido estos datos, comenta: «para entender como será el clima del futuro, debemos entender como ha sido el del pasado». También sucede que la extracción de información a partir de estos datos es costosa, lo que hace necesaria su automatización.

Existen diversos enfoques válidos para resolver este problema de aprendizaje automático. No obstante, nuestro trabajo se basa en el uso de redes neuronales artificiales. El hecho de haber descartado otras técnicas, como las máquinas de vector soporte o la clasificación por vecinos más cercanos, viene motivado por la capacidad que tienen las redes neuronales de obtener buenos resultados sin un preprocesamiento de datos complejo. Esta característica nos permite avanzar con velocidad a una fase de experimentación, donde se valora el uso de técnicas adicionales para la mejora del sistema.

La principal ampliación que implementamos es el uso de un modelo de lenguaje para reconocimiento continuo. El hecho de centrarnos en el reconocimiento de secuencias de caracteres, en lugar de palabras aisladas, viene motivado por el tamaño del vocabulario. Al contar con más de 10.000 palabras diferentes, utilizar un modelo de lenguaje que estimara la probabilidad a priori de cada palabra daría lugar a resultados no competitivos.

Para concluir cabe destacar que, aunque el reconocimiento automático de texto es un área donde se está obteniendo buenos resultados con la tecnología actual, nos encontramos ante una tarea que todavía mantiene el interés académico. La transcripción automática, especialmente de textos históricos, tiene multitud de problemas asociados a los datos para los que todavía no existe una solución estandarizada; tal y como ponemos de manifiesto en nuestra revisión del estado de la cuestión.

[1] Los datos del observatorio Fort William son públicos y están disponibles en: http://brohan.org/OCR-weatherrescue/index.html

1.2 Objetivos

Las diferentes metas a alcanzar con el desarrollo de nuestro trabajo son:

1. Conseguir la correcta extracción de la caja de inclusión de cada fila de los documentos escaneados.
2. Entrenar un modelo neuronal capaz de reconocer los carácteres de las imágenes de manera efectiva.
3. Incluir técnicas de reconocimiento de texto manuscrito derivadas a partir de sistemas de reconocimiento automático del habla para aumentar la precisión del sistema.

1.3 Estructura de la memoria

La estructura de la memoria consiste en cinco apartados principales. Primeramente, se hará una revisión del estado actual del tópico de nuestro trabajo, comentando varias de las técnicas que se utilizan en la actualidad y el ámbito de aplicación de la teconología. En segundo lugar, definiremos brevemente los aspectos teóricos en los que se basa nuestro trabajo y los relacionaremos con las herramientas que se han utilizado durante la implementación. Seguidamente, haremos una descripción exhaustiva de nuestro enfoque a la hora de afrontar este proyecto. En esta sección se incluye la descripción del sistema creado, el preprocesado de las imágenes para el posterior entrenamiento del modelo y las técnicas que se ha utilizado con el objetivo de aumentar la precisión del sistema. La cuarta parte del documento se centra en las pautas seguidas durante la experimentación, así como en la evaluación del rendimiento del sistema siguiendo las métricas estándar. En esta sección se incluye un análisis de los resultados, haciendo hincapié en los distintos tipos de errores que el sistema comete. Posteriormente, revisaremos los objetivos fijados y analizaremos lo logrado con nuestro proyecto. En este capítulo plantearemos algunas opciones que permitirían enriquecer el trabajo realizado pero que no se han llevado a cabo al surgir impedimentos durante su desarrollo.

CAPÍTULO 2
Estado del arte

El reconocimiento automático de texto es un área de interés actual, tanto a nivel académico como de aplicación. Prueba de ello es la diversificación de la materia y la relación que tiene con otras áreas del aprendizaje automático, cuestión que ponemos de manifiesto en este capítulo.

2.1 Uso de técnicas de reconocimiento del habla

Aunque el reconocimiento de texto y el reconocimiento del habla son áreas del aprendizaje automático distintas, en cuanto a la fuente de información que utilizan para extraer sus resultados, las técnicas que se utilizan a la hora de inferir resultados suelen coincidir. Por tanto, es normal que los avances en investigación del reconocimiento del habla sean también avances para el reconocimiento de texto.

Un ejemplo del uso de la tecnología de reconocimiento del habla lo encontramos en [6], donde se utiliza un modelo de lenguaje de n-grama basado en caracteres para incluir información contextual en el proceso de clasificación. El uso de un modelo de lenguaje es una técnica popular entre los trabajos actuales sobre reconocimiento de texto, técnica que explicamos en mayor detalle en el siguiente capítulo.

2.2 Reconocimiento de textos históricos

Existen múltiples ejemplos donde la transcripción automática se ha aplicado a escrituras antiguas. En [3] se discute el rendimiento de distintos modelos, sometidos a competición, a la hora de lidiar con la transcripción de manuscritos científicos antiguos escritos en árabe. La tarea se divide en tres partes: segmentación de páginas, extracción de las líneas del texto y reconocimiento óptico de caracteres.

Uno de los competidores se basa en Google Cloud Vision [16], que utiliza redes neuronales convolucionales para la detección de líneas en el texto. Después se aplican una serie de heurísticas para determinar la dirección de escritura y el estilo de escritura, haciendo que el sistema de reconocimiento de texto, también basado en redes neuronales convolucionales, esté más informado.

Los principales problemas a los que se enfrenta la transcripción automática de manuscritos antiguos son la baja calidad de la imagen de muestra, al haberse deteriorado los textos con el paso del tiempo, y la falta de muestras etiquetadas. Ha de tenerse en cuenta que la transcripción manual de estos textos es costosa, lo que limita en gran manera los métodos de aprendizaje supervisado.

En [2] hacen frente a los problemas expuestos haciendo uso del aprendizaje por transferencia de una red generativa antagónica. De esta manera, se parte de un modelo preentrenado sobre el cual se hace una ligera cantidad de iteraciones sobre el reducido conjunto de datos. Gracias al uso del aprendizaje por transferencia se aumenta, en cierto modo, el tamaño del conjunto de datos de entrenamiento.

El modelo consta de dos componentes: el generador y el discriminador. En la solución propuesta, ambos componentes se basan en el uso de redes neuronales convolucionales. El generador trata de, a partir de las imágenes de entrada, engañar al discriminador produciendo imágenes incorrectas que se asemejen a las reales. El discriminador aprende a discernir entre imágenes válidas y aquellas inválidas producidas, de tal manera que se entrena para reconocer la caligrafía de cada carácter de la época frente a posibles copias.

Sin embargo, el aprendizaje por transferencia no es la única opción válida para atacar esta problemática. En [10] se hace uso de una red neuronal convolucional recurrente[1] para la lectura de manuscritos en tibetano. Esta red se entrena a partir de datos sintéticos, generados a partir de las muestras de entrada, sobre los que se añade ruido artificial. El modelo se adapta, posteriormente, a las muestras reales siguiendo un proceso de regularización.

A la hora de evaluar este sistema se utilizan las muestras etiquetadas originales. Como los datos generados artificialmente consisten en líneas completas y las muestras son páginas con su transcripción, es necesario un proceso de segementación por líneas. El modelo encargado de realizar este proceso aprende sus parámetros utilizando métodos de aprendizaje no supervisado.

Los segmentadores de líneas acostumbran a aprender sus parámetros a partir de unas muestras, ya sea utilizando métodos supervisados o sin supervisión. En [9] se presenta un método de extracción de líneas que no aprende sus parámetros, sino que utiliza un grafo de distancias como proyección del documento y trabaja sobre dicho grafo para conseguir la segmentación.

Otro problema propio de los documentos antiguos es la cantidad de texto corrido que aparece. A la hora de aprender a diferenciar los distintos caracteres que aparecen en el texto es vital contar con muestras aisladas de cada carácter. En [17] hacen uso de redes convolucionales completas, además de una fase de posprocesado, para realizar la segmentación por carácteres del texto japonés.

2.3 Reconocimiento de texto en línea

La transcripción automática no se limita a tareas *offline*, donde la entrada viene en forma de imágenes escaneadas. Los sistemas de reconocimiento automático de texto en línea han de tener en cuenta la velocidad con la que se procesan los datos, además de la precisión en la transcripción. Sin embargo, poseen información sobre el dominio temporal de la señal que los sistemas *offline* han de simular.

En [7] se expone el uso de redes neuronales convolucionales profundas para el reconocimiento en línea de carácteres manuscritos, así como la posible aplicación práctica de la tecnología y la distinción entre las distintas áreas del reconocimiento de texto. También incide en la información que se puede extraer de los datos para enriquecer el proceso de inferencia, como la presión a la hora de escribir o la velocidad del trazo.

[1] Una red convolucional recurrente consiste en una secuencia de capas convolucionales que implementan un procesamiento de la imagen para producir la entrada de la red recurrente, cuestión que explicamos en mayor detalle en el capítulo 4.

2.4 Reconocimiento de texto en escenas

Otra posible aplicación de la transcripción automática es integrarla en sistemas de análisis de imágen. El objetivo del reconocimiento de texto en escenas es el de detectar secciones de las imágenes que puedan contener texto escrito y hacer su transcripción. En este caso, el énfasis recae más sobre el tratamiento de la imagen que sobre la propia transcripción, aunque también juega un papel fundamental para la consecución de este objetivo.

En [4] se evalúa el rendimiento de distintos sistemas en un total de 3 tareas de respuesta a preguntas basado en imágenes, ordenadas por dificultad ascendente. Esta competición es de las primeras en incoporar texto a esta tarea, y abre una nueva área de aplicación para el mundo de la transcripción automática.

CAPÍTULO 3
Fundamentos teóricos

En este capítulo definiremos algunos de los conceptos fundamentales para entender nuestra solución y nombraremos las herramientas que se han utilizado para aplicar dichos conceptos.

3.1 Redes neuronales artificiales

Las redes neuronales artificiales se utilizan tanto para problemas de clasificación, donde se pretende agrupar las muestras en función de sus características, como de regresión, donde se quiere codificar las muestras. El elemento fundamental de estas redes es la neurona, la cual calcula una función de la entrada haciendo uso de un vector de pesos que ha de ajustarse durante la fase de entrenamiento.

La arquitectura de una red neuronal artificial suele consistir en una serie de capas de neuronas, donde las neuronas de una capa reciben como entrada las salidas de la capa previa. Las muestras se suministran a través de la capa de entrada, en forma de vector numérico, y la salida de la red se lee como los resultados de la última capa. Para saber más acerca de los componentes de una red neuronal y de su arquitectura recomendamos leer el capítulo 3 de [12].

El sistema ajusta el vector de pesos de cada neurona a partir de unas muestras y la salida que se espera, para ser capaz de reconocer adecuadamente cada muestra. Los fundamentos del entrenamiento de redes neuronales artificiales y los algoritmos que se utilizan se pueden consultar en el capítulo 5 de [12]. Cabe comentar, no obstante, que los métodos más populares en la actualidad se basan en la minimización, por descenso por gradiente, de una función de error.

Para la construcción y entrenamiento de una red neuronal artificial que clasifique las muestras de nuestro problema, hacemos uso de PyLaia [13]: un conjunto de herramientas de alto nivel que nos permite generar modelos con los parámetros que especifiquemos sin entrar en detalles de implementación de la propia red. PyLaia aprovecha la potencia de cálculo matricial de la *Graphical Processor Unit* (GPU) para acelerar el proceso de entrenamiento de la red, esto lo hace a través de la librería CUDA de NVIDIA. Gracias a este aumento de rendimiento, podemos obtener resultados en un tiempo mucho menor al que necesitaríamos si utilizáramos únicamente el procesador de nuestra máquina.

3.2 Modelos ocultos de Márkov

Un modelo oculto de Márkov [15] es un formalismo matemático que representa un proceso doblemente estocástico en el que uno de los procesos es desconocido y se puede conocer a través de la observación sistemática de resultados. El proceso es doblemente estocástico porque ocurren dos eventos guiados por la probabilidad: la transición de un estado a otro y la generación de símbolos en cada estado.

De esta manera, podemos definir un modelo oculto de Márkov mediante un grafo donde los nodos representan estados, con una probabilidad de emisión definida para cada símbolo, y los arcos representan la probabilidad de transición de un estado a otro. La figura 3.1 trata de representar gráficamente un posible modelo oculto de Márkov.

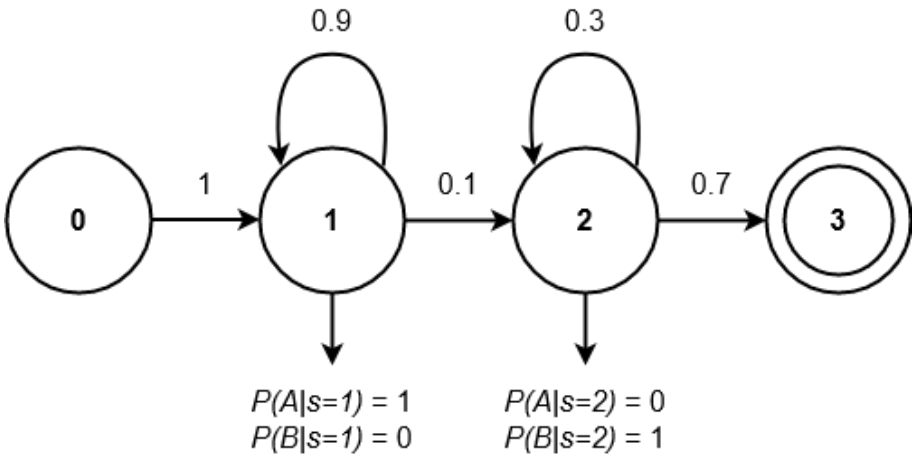


Figura 3.1: Representación gráfica de un modelo oculto de Márkov.

Aunque los modelos ocultos de Markov tienen una gran cantidad de aplicaciones en el mundo del reconocimiento de formas y del aprendizaje automático, en nuestro caso estamos interesados en dos de ellas. Primero de todo, estamos interesados en la existencia de algoritmos eficientes para estimar automáticamente los parámetros del modelo a partir de las muestras que se esperan de salida. También nos interesa, con un modelo ya entrenado, ser capaces de calcular la probabilidad de que se dé una secuencia de salida con un algoritmo eficiente en tiempo.

3.3 Modelo de lenguaje de n-grama

Un modelo de lenguaje permite predecir qué palabra debería suceder al texto generado en un sistema. Esto lo consigue teniendo en cuenta las palabras que se han generado previamente, es decir: analizando el contexto. La predicción se formula como una probabilidad condicional para cada palabra del vocabulario: la probabilidad de que dicha palabra sea la siguiente en aparecer dado el contexto. Es decir: $P(w) = P(w|c)$

El modelo de lenguaje de n-grama, explicado en gran detalle en las páginas dos a siete del capítulo 3 de [8], es uno de los más sencillos. En esencia, asigna la probabilidad condicional teniendo en cuenta las $n-1$ palabras que hayan aparecido anteriormente. Como se puede observar, a mayor n mayor es la longitud del contexto a tener en cuenta y, por tanto, más complejo es el modelo. La probabilidad que aproxima este modelo es: $P(w|c) = P(w|w_0 w_1 .. w_{n-1})$

Dicho modelo de lenguaje de n-grama ve su implementación, en realidad, como un *Stochastic Finite State Automata* (SFSA). Para la creación y entrenamiento de los modelos de lenguaje haremos uso del *toolkit* de reconocimiento del habla "Kaldi".[1]

La incorporación de un modelo de lenguaje de n-grama hace que el sistema no solo tenga en cuenta el componente óptico, la red neuronal artificial, para asignar la probabilidad a una hipótesis. Para controlar que la aportación de ambos modelos sea equilibrada o, mejor dicho, la óptima, se ajusta un hiperparámetro del sistema conocido como el *Grammage Scale Factor* (GSF).

Al trabajar con modelos de lenguaje, ocurre también que las hipótesis que más peso tienen son aquellas con una longitud más corta. Esto se debe a que, para el cálculo de la probabilidad a priori de una hipótesis, se suele emplear un productorio de probabilidades que van de cero a uno. Por tanto, a mayor longitud de la hipótesis menor es dicha probabilidad.

La *Word Insertion Penalty* (WIP) es un hiperparámetro del sistema que permite controlar el efecto descrito al priorizar las hipótesis más largas. Dicho hiperparámetro ha de ser ajustado con un conjunto de validación, similar a como ocurre con el GSF. Teniendo ambos parámetros en cuenta, así como el hecho de que se va a utilizar un modelo óptico, la secuencia de palabras con mayor probabilidad de generación a partir de la muestra x es:

$$\begin{aligned} \bar{w} = \arg\max_{w} P(w|x) &= arg\max_{w} \frac{P(w,x)}{P(x)} = \arg\max_{w} P(x|w)\,P(w) \\ &= \arg\max_{w} \log P(x|w) + \alpha \log P(w) - n\,Q \end{aligned}$$

Donde n es la longitud de la secuencia de palabras, Q es la WIP y α es el GSF. Además, $\log P(x|w)$ viene dado por el modelo óptico y $\log P(w)$ por el modelo de lenguaje.

3.4 Búsqueda en haz y *lattices*

A la hora de calcular la transcripción más probable a partir de la señal de entrada, se tiene que hacer un análisis de izquierda a derecha de dicha entrada teniendo en cuenta las posibles salidas del sistema. Sin embargo, cuando se trabaja con un número potencialmente infinito de hipótesis, obtener la transcripción se convierte en un problema algorítmico que requiere una solución eficiente. Hay que tener en cuenta que la solución óptima no se puede encontrar haciendo una búsqueda exhaustiva.

La búsqueda en haz[2] es un algoritmo de exploración que somete a poda las hipótesis consideradas por el sistema durante la exploración. Este algoritmo sacrifica la optimalidad de la solución al descartar caminos que podrían alcanzar la solución óptima en favor de mantener un conjunto más reducido de hipótesis. No obstante, en la práctica ha demostrado dar buenos resultados.

Conociendo la hipóteis más probable podemos, con un sobrecoste mínimo, conocer las n hipótesis más probables y la secuencia de estados que las alcanzan. Este conjunto de hipótesis se condensa en un grafo, conocido como *lattice*.

[1] Documentación de Kaldi: http://kaldi-asr.org/doc/

[2] Para una explicación más detallada del algoritmo de búsqueda en haz, el lector puede visitar: https://medium.com/@dhartidhami/beam-search-in-seq2seq-model-7606d55b21a5

3.5 Validación cruzada

Existen múltiples métodos[3] para la evaluación de la precisión de un sistema de reconocimiento como el nuestro. La opción más sencilla consiste en dejar un subconjunto de prueba para evaluar la capacidad del sistema, que se entrena con el resto de datos. Al variar el rendimiento del sistema en función del conjunto de datos escogido, se podría decir que esta técnica no es lo suficientemente rigurosa.

La validación cruzada da un paso más, entrenando k sistemas a partir de las k divisiones que se hacen sobre el conjunto de datos original. Esto es, existen k conjuntos de datos de entrenamiento y k conjuntos de datos de test. El rendimiento del sistema global se calcula, entonces, como la media aritmética del rendimiento de cada uno de los k sistemas. En [11] se evalúa diferentes métodos de selección de modelos y se termina concluyendo que usar validación cruzada con 10 particiones de igual tamaño es la mejor alternativa, motivo por el que nos aproximaremos lo máximo posible a este número de divisiones.

3.6 *Character Error Rate* y *Word Error Rate*

Las métricas a minimizar con el entrenamiento del sistema son el *Character Error Rate* (CER) y el *Word Error Rate* (WER), ambas calculadas mediante los métodos que nos ofrece PyLaia. El CER resulta de dividir la distancia de Levenshtein,[4] del texto de salida al texto objetivo, entre el número de carácteres del texto que se quiere obtener. De esta forma, el CER mide el índice de error a nivel de carácter que se puede encontrar en la transcripción.

El cálculo del WER es similar al del CER, excepto que en este caso los errores se miden como la inserción, borrado o sustitución de palabras completas. La distancia del texto de salida al texto objetivo se divide entre el número de palabras del texto, resultando en el índice de error a nivel de palabra que hay en el texto de salida.

La propia definición de estas métricas nos da a entender que el WER es más pesimista que el CER, ya que un error en un solo carácter de la palabra hace que la palabra completa sea incorrecta. La prueba de ello la veremos con la naturaleza de los errores y la evaluación del rendimiento del sistema, en el capítulo 5 de este documento.

[3] Para una explicación más detallada sobre los métodos de validación de modelos estadísticos recomendamos leer: https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap

[4] Más información sobre el cálculo de la distancia de Levenshtein en: https://dzone.com/articles/the-levenshtein-algorithm-1

CAPÍTULO 4
Desarrollo de la solución

En este capítulo comentamos los distintos módulos que componen nuestra solución, así como el análisis previo que hemos hecho de la tarea a resolver y los métodos de preprocesamiento de datos que hemos empleado.

4.1 Análisis del problema

El objetivo de nuestra tarea es automatizar la transcripción de documentos escaneados en formato tabla. Para ello, a partir de una imagen, se tiene que hacer un proceso para obtener un archivo en formato *Comma-Separated Values* (CSV) con la información.

Los datos, cortesía de la iniciativa Old Weather, consisten en un conjunto de imágenes de texto impreso donde cada imagen representa la evolución de la presión atmosférica en el observatorio *Fort-William*. Cada fila de las tablas representa un día y cada columna una hora, además de la fila y la columna destinadas al cálculo de las medias. Si se lee una fila se ve la evolución de la presión durante el día, además de la presión media en dicho día. En la figura 4.1 observamos una de las tablas de las que disponemos.

FORT-WILLIAM OBSERVATORY. 95

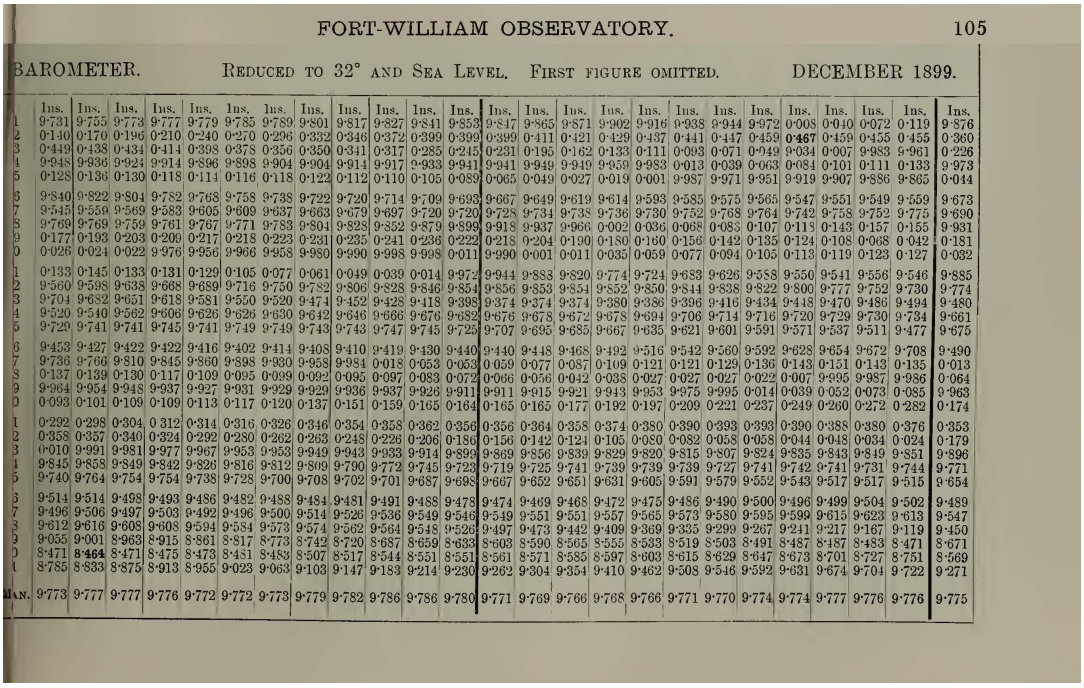
BAROMETER. REDUCED TO 32° AND SEA LEVEL. FIRST FIGURE OMITTED. JANUARY 1899.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Mid-night.	Mean.
	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.
1	9·197	9·191	9·187	9·171	9·149	9·131	9·111	9·091	9·075	9·062	9·036	8·994	8·962	8·926	8·894	8·866	8·852	8·822	8·792	8·778	8·742	8·722	8·700	8·656	8·963
2	8·626	8·590	8·578	8·580	8·572	8·552	8·554	8·570	8·606	8·643	8·673	8·721	8·781	8·828	8·880	8·945	8·998	9·061	9·116	9·172	9·222	9·278	9·325	9·368	8·843
3	9·405	9·457	9·501	9·537	9·573	9·597	9·623	9·647	9·673	9·675	9·676	9·657	9·637	9·614	9·578	9·570	9·515	9·512	9·506	9·510	9·508	9·550	9·600	9·609	9·572
4	9·650	9·669	9·696	9·704	9·714	9·712	9·710	9·711	9·693	9·676	9·611	9·545	9·507	9·489	9·485	9·504	9·485	9·459	9·485	9·545	9·644	9·717	9·775	9·819	9·625
5	9·846	9·862	9·874	9·864	9·858	9·830	9·826	9·907	9·973	0·032	0·059	0·079	0·087	0·117	0·133	0·187	0·207	0·230	0·252	0·275	0·277	0·279	0·266	0·250	0·065
6	0·227	0·219	0·199	0·141	0·089	0·015	9·977	9·939	9·863	9·799	9·771	9·783	9·805	9·885	9·905	9·921	9·937	9·941	9·939	9·945	9·964	9·969	9·965	9·965	9·965
7	9·953	9·947	9·946	9·921	9·911	9·895	9·880	9·871	9·871	9·861	9·850	9·820	9·788	9·756	9·738	9·717	9·700	9·680	9·662	9·653	9·642	9·650	9·647	9·646	9·792
8	9·645	9·639	9·621	9·607	9·600	9·586	9·586	9·599	9·605	9·598	9·596	9·590	9·576	9·560	9·538	9·515	9·491	9·480	9·469	9·456	9·451	9·430	9·390	9·370	9·542
9	9·332	9·317	9·301	9·279	9·270	9·287	9·301	9·316	9·339	9·348	9·349	9·329	9·317	9·293	9·266	9·249	9·233	9·205	9·195	9·155	9·147	9·130	9·106	9·081	9·255
0	9·061	9·037	9·007	8·993	8·969	8·964	8·969	8·981	8·981	8·991	9·008	9·000	9·016	9·024	9·044	9·052	9·066	9·090	9·106	9·124	9·132	9·149	9·149	9·142	9·044
1	9·138	9·146	9·151	9·151	9·150	9·150	9·154	9·158	9·177	9·188	9·180	9·159	9·141	9·150	9·159	9·165	9·166	9·182	9·196	9·214	9·228	9·236	9·243	9·236	9·175
2	9·230	9·218	9·194	9·144	9·084	9·024	8·956	8·886	8·792	8·712	8·655	8·623	8·606	8·603	8·631	8·713	8·795	8·906	9·015	9·103	9·180	9·239	9·291	9·331	8·955
3	9·366	9·402	9·414	9·420	9·428	9·432	9·430	9·427	9·424	9·397	9·404	9·388	9·368	9·375	9·380	9·381	9·388	9·402	9·419	9·412	9·434	9·458	9·471	9·495	9·413
4	9·508	9·529	9·538	9·554	9·574	9·591	9·618	9·649	9·668	9·681	9·692	9·689	9·679	9·673	9·657	9·653	9·653	9·649	9·657	9·667	9·679	9·687	9·699	9·717	9·640
5	9·723	9·739	9·735	9·723	9·732	9·727	9·720	9·699	9·660	9·630	9·618	9·590	9·536	9·510	9·480	9·446	9·360	9·307	9·267	9·200	9·162	9·137	9·085	9·076	9·494
6	9·036	9·029	9·003	8·963	8·931	8·936	8·923	8·927	8·926	8·955	8·977	8·987	9·017	9·051	9·085	9·143	9·191	9·245	9·299	9·368	9·431	9·483	9·527	9·582	9·126
7	9·630	9·656	9·702	9·722	9·742	9·769	9·791	9·812	9·838	9·854	9·858	9·866	9·856	9·856	9·858	9·854	9·852	9·838	9·830	9·819	9·796	9·776	9·735	9·708	9·792
8	9·655	9·608	9·580	9·560	9·506	9·447	9·410	9·392	9·367	9·343	9·316	9·280	9·248	9·216	9·187	9·156	9·124	9·116	9·096	9·092	9·104	9·122	9·144	9·156	9·301
9	9·155	9·149	9·143	9·126	9·110	9·093	9·085	9·103	9·100	9·114	9·135	9·150	9·154	9·156	9·166	9·162	9·156	9·138	9·128	9·120	9·123	9·094	9·080	9·062	9·125
0	9·045	9·026	9·044	9·096	9·162	9·201	9·241	9·278	9·303	9·320	9·351	9·349	9·337	9·331	9·327	9·325	9·318	9·307	9·281	9·268	9·253	9·219	9·176	9·149	9·238
1	9·127	9·083	9·053	9·017	8·965	8·959	8·938	8·947	8·961	8·955	8·968	8·968	8·960	8·949	8·946	8·954	8·958	8·964	8·964	8·960	8·952	8·941	8·936	8·926	8·973
2	8·924	8·924	8·920	8·922	8·922	8·925	8·938	8·968	9·000	9·052	9·091	9·123	9·155	9·189	9·225	9·271	9·315	9·361	9·410	9·463	9·512	9·565	9·607	9·647	9·185
3	9·694	9·738	9·770	9·796	9·825	9·863	9·897	9·933	9·973	0·016	0·038	0·065	0·086	0·100	0·122	0·144	0·174	0·204	0·238	0·266	0·269	0·287	0·317	0·320	0·047
4	0·321	0·333	0·339	0·351	0·363	0·373	0·385	0·404	0·421	0·441	0·449	0·455	0·456	0·465	0·465	0·473	0·469	0·475	0·485	0·499	0·509	0·518	0·523	0·527	0·437
5	0·535	0·543	0·548	0·549	0·549	0·549	0·560	0·573	0·591	0·593	0·606	0·590	0·588	0·574	0·575	0·577	0·592	0·601	0·616	0·621	0·625	0·640	0·651	0·650	0·587
6	0·656	0·656	0·656	0·652	0·651	0·647	0·649	0·658	0·674	0·677	0·677	0·675	0·662	0·654	0·659	0·653	0·658	0·654	0·647	0·651	0·647	0·646	0·635	0·635	0·655
7	0·626	0·615	0·617	0·608	0·607	0·599	0·596	0·597	0·603	0·599	0·601	0·592	0·574	0·560	0·542	0·542	0·536	0·538	0·534	0·529	0·530	0·538	0·520	0·514	0·572
8	0·510	0·507	0·493	0·483	0·468	0·465	0·458	0·464	0·467	0·471	0·468	0·440	0·429	0·399	0·399	0·391	0·389	0·377	0·365	0·351	0·357	0·339	0·333	0·321	0·423
9	0·309	0·297	0·291	0·283	0·273	0·263	0·260	0·263	0·263	0·249	0·240	0·220	0·206	0·186	0·174	0·177	0·184	0·183	0·176	0·176	0·169	0·161	0·154	0·150	0·221
0	0·144	0·136	0·133	0·124	0·116	0·112	0·110	0·118	0·127	0·126	0·122	0·110	0·104	0·086	0·070	0·071	0·079	0·068	0·062	0·054	0·044	0·012	0·002	9·969	0·087
1	9·960	9·944	9·922	9 894	9·874	9·846	9·827	9·806	9·794	9·784	9·767	9·740	9·716	9·688	9·666	9·650	9·638	9·624	9·611	9·604	9·602	9·598	9·593	9·586	9·739
MAN.	9·653	9·652	9·650	9·643	9·637	9·630	9·628	9·635	9·639	9·640	9·640	9·632	9·624	9·621	9·620	9·627	9·628	9·633	9·639	9·647	9·656	9·664	9·666	9·667	9·640

Figura 4.1: Datos recogidos durante el mes de enero de 1899.

Como se puede apreciar, los carácteres están escritos a máquina y los espacios entre columnas y entre filas son bastante regulares. Estos factores hacen que el conjunto de datos sea muy regular, lo cual simplifica el proceso de aprendizaje y le resta importancia al uso de un modelo de lenguaje.

Sin embargo, hay pequeñas irregularidades en las imágenes que cabe comentar. En la figura 4.2 podemos observar como la tabla tiene una disposición ligeramente diferente a la de los datos del mes de enero. Esto es porte en los registros de enero se incluye una pequeña cabecera que indica la hora a la que se refiere cada columna. Dicha cabecera aparece en todos los meses impares del año.



FORT-WILLIAM OBSERVATORY. 105

BAROMETER. REDUCED TO 32° AND SEA LEVEL. FIRST FIGURE OMITTED. DECEMBER 1899.

	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.
1	9·731	9·755	9·773	9·777	9·779	9·785	9·789	9·801	9·817	9·827	9·841	9·853	9·847	9·865	9·871	9·902	9·916	9·938	9·944	9·972	0·008	0·040	0·072	0·119	9·876
2	0·140	0·170	0·196	0·210	0·240	0·270	0·298	0·332	0·346	0·372	0·399	0·399	0·399	0·411	0·421	0·429	0·437	0·441	0·447	0·459	0·467	0·459	0·455	0·455	0·360
3	0·449	0·438	0·434	0·414	0·398	0·378	0·356	0·350	0·341	0·317	0·285	0·245	0·231	0·195	0·162	0·133	0·111	0·093	0·071	0·049	9·034	9·007	9·983	9·961	0·226
4	9·948	9·936	9·924	9·914	9·896	9·898	9·904	9·904	9·914	9·917	9·933	9·941	9·941	9·949	9·949	9·959	9·983	0·013	0·039	0·063	0·084	0·101	0·111	0·133	9·973
5	0·128	0·136	0·130	0·118	0·114	0·116	0·118	0·122	0·112	0·110	0·105	0·089	0·065	0·049	0·027	0·019	0·001	9·987	9·971	9·951	9·919	9·907	9·886	9·865	0·044
6	9·840	9·822	9·804	9·782	9·768	9·758	9·738	9·722	9·720	9·714	9·709	9·693	9·667	9·649	9·619	9·614	9·593	9·585	9·575	9·565	9·547	9·551	9·549	9·559	9·673
7	9·545	9·559	9·569	9·583	9·605	9·609	9·637	9·663	9·679	9·697	9·720	9·720	9·728	9·734	9·738	9·736	9·730	9·752	9·768	9·764	9·742	9·758	9·752	9·775	9·690
8	9·769	9·769	9·759	9·761	9·767	9·771	9·783	9·804	9·828	9·852	9·879	9·899	9·918	9·937	9·966	0·002	0·036	0·068	0·083	0·107	0·118	0·143	0·157	0·155	9·931
9	0·177	0·193	0·203	0·209	0·217	0·218	0·223	0·231	0·235	0·241	0·236	0·222	0·218	0·204	0·190	0·180	0·160	0·156	0·142	0·135	0·124	0·108	0·068	0·042	0·181
0	0·026	0·024	0·022	9·976	9·956	9·966	9·958	9·980	9·990	9·998	9·998	0·011	9·990	0·001	0·011	0·035	0·059	0·077	0·094	0·105	0·113	0·119	0·123	0·127	0·032
1	0·133	0·145	0·133	0·131	0·129	0·105	0·077	0·061	0·049	0·039	0·014	9·972	9·944	9·883	9·820	9·774	9·724	9·683	9·626	9·588	9·550	9·541	9·556	9·546	9·885
2	9·560	9·598	9·638	9·668	9·689	9·716	9·750	9·782	9·806	9·828	9·846	9·854	9·856	9·853	9·854	9·852	9·850	9·844	9·838	9·822	9·800	9·777	9·752	9·730	9·774
3	9·704	9·682	9·651	9·618	9·581	9·550	9·520	9·474	9·452	9·428	9·418	9·398	9·374	9·374	9·374	9·380	9·386	9·396	9·416	9·434	9·448	9·470	9·486	9·494	9·480
4	9·520	9·540	9·562	9·606	9·626	9·626	9·630	9·642	9·646	9·666	9·676	9·682	9·676	9·678	9·672	9·678	9·694	9·706	9·714	9·716	9·720	9·729	9·730	9·734	9·661
5	9·729	9·741	9·741	9·745	9·741	9·749	9·749	9·743	9·743	9·747	9·745	9·725	9·707	9·695	9·685	9·667	9·635	9·621	9·601	9·591	9·571	9·537	9·511	9·477	9·675
6	9·453	9·427	9·422	9·422	9·416	9·402	9·414	9·408	9·410	9·419	9·430	9·440	9·440	9·448	9·468	9·492	9·516	9·542	9·560	9·592	9·628	9·654	9·672	9·708	9·490
7	9·736	9·766	9·810	9·845	9·860	9·898	9·930	9·958	9·984	0·018	0·053	0·053	0·059	0·077	0·087	0·109	0·121	0·121	0·129	0·136	0·143	0·151	0·143	0·135	0·013
8	0·137	0·139	0·130	0·117	0·109	0·095	0·099	0·092	0·095	0·097	0·083	0·072	0·066	0·056	0·042	0·038	0·027	0·027	0·027	0·022	0·007	9·995	9·987	9·986	0·064
9	9·964	9·954	9·948	9·937	9·927	9·931	9·929	9·929	9·936	9·937	9·926	9·911	9·911	9·915	9·921	9·943	9·953	9·975	9·995	0·014	0·039	0·052	0·073	0·085	9·963
0	0·093	0·101	0·109	0·109	0·113	0·117	0·120	0·137	0·151	0·159	0·165	0·164	0·165	0·165	0·177	0·192	0·197	0·209	0·221	0·237	0·249	0·260	0·272	0·282	0·174
1	0·292	0·298	0·304	0 312	0·314	0·316	0·326	0·346	0·354	0·358	0·362	0·356	0·356	0·364	0·358	0·374	0·380	0·390	0·393	0·393	0·390	0·388	0·380	0·376	0·353
2	0·358	0·357	0·340	0·324	0·292	0·280	0·262	0·263	0·248	0·226	0·206	0·186	0·156	0·142	0·124	0·105	0·080	0·082	0·058	0·058	0·044	0·048	0·034	0·024	0·179
3	0·010	9·991	9·981	9·977	9·967	9·953	9·953	9·949	9·943	9·933	9·914	9·899	9·869	9·856	9·839	9·829	9·820	9·815	9·807	9·824	9·835	9·843	9·849	9·851	9·896
4	9·845	9·858	9·849	9·842	9·826	9·816	9·812	9·809	9·790	9·772	9·745	9·723	9·719	9·725	9·741	9·739	9·739	9·739	9·727	9·741	9·742	9·741	9·731	9·744	9·771
5	9·740	9·764	9·754	9·754	9·738	9·728	9·700	9·708	9·702	9·701	9·687	9·698	9·667	9·652	9·651	9·631	9·605	9·591	9·579	9·552	9·543	9·517	9·517	9·515	9·654
6	9·514	9·514	9·498	9·493	9·486	9·482	9·488	9·484	9·481	9·491	9·488	9·478	9·474	9·469	9·468	9·472	9·475	9·486	9·490	9·500	9·496	9·499	9·504	9·502	9·489
7	9·496	9·506	9·497	9·503	9·492	9·496	9·500	9·514	9·526	9·536	9·549	9·546	9·549	9·551	9·551	9·557	9·565	9·573	9·580	9·595	9·599	9·615	9·623	9·613	9·547
8	9·612	9·616	9·608	9·608	9·594	9·584	9·573	9·574	9·562	9·564	9·548	9·526	9·497	9·473	9·442	9·409	9·369	9·335	9·299	9·267	9·241	9·217	9·167	9·119	9·450
9	9·055	9·001	8·963	8·915	8·861	8·817	8·773	8·742	8·720	8·687	8·659	8·633	8·603	8·590	8·565	8·555	8·533	8·519	8·503	8·491	8·487	8·487	8·483	8·471	8·671
0	8·471	8·464	8·471	8·475	8·473	8·481	8·483	8·507	8·517	8·544	8·551	8·551	8·561	8·571	8·585	8·597	8·603	8·615	8·629	8·647	8·673	8·701	8·727	8·751	8·569
1	8·785	8·833	8·875	8·913	8·955	9·023	9·063	9·103	9·147	9·183	9·214	9·230	9·262	9·304	9·354	9·410	9·462	9·508	9·546	9·592	9·631	9·674	9·704	9·722	9·271
MAN.	9·773	9·777	9·777	9·776	9·772	9·772	9·773	9·779	9·782	9·786	9·786	9·780	9·771	9·769	9·766	9·768	9·766	9·771	9·770	9·774	9·774	9·777	9·776	9·776	9·775

Figura 4.2: Datos recogidos durante el mes de diciembre de 1899.

Hay otros factores que pueden mermar los resultados si no se tienen en cuenta. Algunas páginas, por la manera en la que se han escaneado, aparecen con una ligera inclinación o curvatura que pueden complicar la extracción de las líneas. En Análisis de la Maquetación de Documentos, conocido en inglés como *Document Layout Analysis*, estos problemas se identifican, en inglés, como *skew* y *warping*.

Otro de los problemas al que tenemos que hacer frente es la presencia de números incompletos o incluso la falta del contenido de alguna celda. Este caso, aunque poco común, suele darse en las tablas que presentan una curvatura pronunciada. En esta situación es muy difícil que el modelo óptico sea capaz de obtener alguna transcripción, por lo que no se esperan buenos resultados en estos casos.

También cabe comentar que los únicos datos que nos interesan son aquellos que se encuentran dentro de la tabla, por lo que tendremos que descartar la cabecera común a todas las páginas. Este será uno de los objetivos a resolver durante el procesamiento de los datos previo al entrenamiento del modelo.

En líneas generales, estamos ante un problema de reconocimiento *offline* de carácteres escritos a máquina. La manera de abordar este problema es similar a la de una tarea de reconocimiento de texto manuscrito. En nuestro caso, hacemos uso de las redes neuronales artificiales. Otros métodos válidos, aunque quizás menos competitivos, son la clasificación por vecinos más cercanos o el uso de máquinas de vectores soporte.

4.2 Procesamiento de los datos

Para poder trabajar con el modelo de reconocimiento se le ha de presentar los datos en el formato adecuado. En nuestro caso separamos las tablas por filas, es decir: por días. Para conseguir este objetivo originalmente probamos un método bastante básico que se basaba en la obtención de máximos relativos en un histograma obtenido a partir de la proyección en el eje vertical de los píxeles negros en una versión binarizada de la imagen.

Con un poco más de detalle, primero se hacía un conteo por filas del número de píxeles cuya componente roja estaba por encima de un límite. Dicho límite se fijó tras observar los valores de intensidad de los píxeles de fondo y los carácteres de la tabla. Después, se recorría el histograma para encontrar los máximos relativos, que se correspondían con filas donde una gran mayoría de los píxeles estaban en blanco. Con esta información, se podía extraer las cajas que correspondían a un registro de la tabla: espacio entre dos líneas en blanco. Este método resultó dar resultados poco acertados al ser especialmente sensible a la inclinación de las páginas, ya que trazaba líneas horizontales para la división tal y como se puede observar en la figura 4.3.

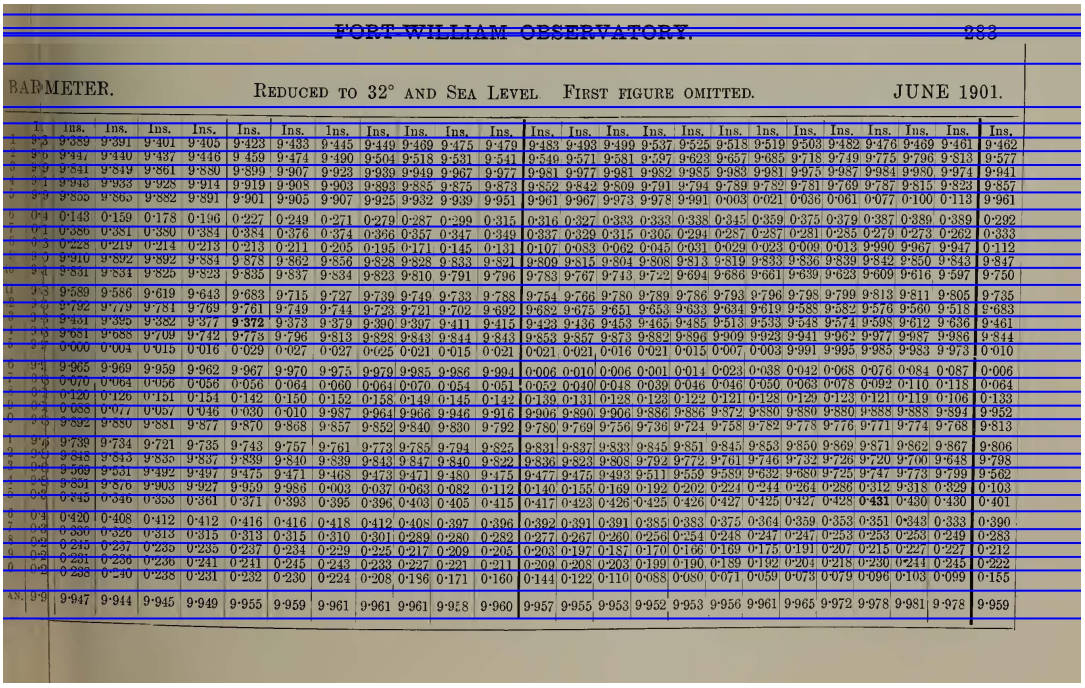


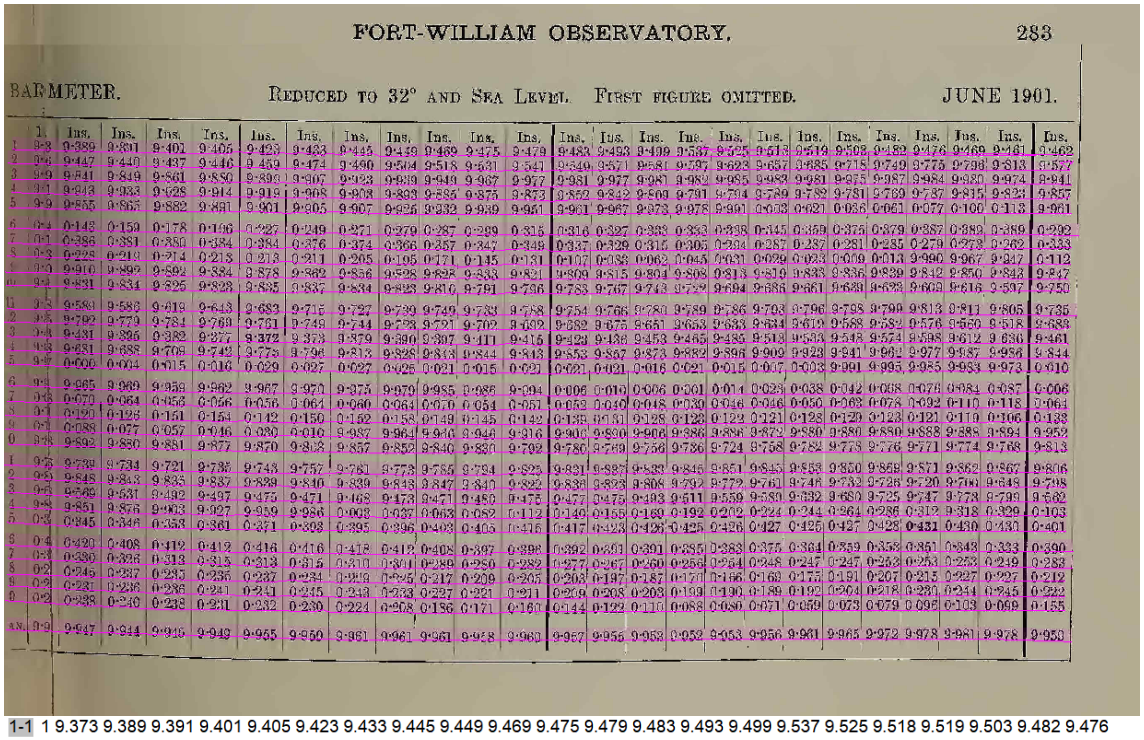
Figura 4.3: Resultado de la división de la imagen utilizando el método original. Nótese como en las últimas líneas el sistema tiene problemas dada la curvatura.

Un método más robusto para la resolución de este problema lo encontramos en [14], donde se utiliza redes neuronales artificiales para un análisis completo del documento; incluyendo la geometría y disposición del mismo. Gracias a este trabajo podemos extraer con seguridad la información de las páginas que tienen curvatura.

Con los resultados de dicho modelo solo resta eliminar la información superflua, principalmente los elementos de la cabecera; hacer un retoque manual de las líneas para asegurar que incluyen todos los elementos de la línea y añadir la información del *ground truth* a cada una de las muestras. Este último preprocesamiento lo hacemos de manera manual con Transkribus,[1] un software dedicado a la transcripción de documentos que nos permite exportar los archivos en el formato que requiere nuestro sistema.

[1]Página web de Transkribus: https://transkribus.eu/Transkribus/

En la sección de experimentación de resultados, no obstante, probaremos el rendimiento del sistema con la modificación manual del tamaño de las líneas y sin dicha modificación. En la figura 4.4 mostramos el resultado de la extracción automática con redes neuronales y el retoque manual tal y como se ve en Transkribus. En la figura 4.5 mostramos algunas de las líneas, extraídas de la misma imagen, en el formato que recibirá el sistema.



1-1 1 9.373 9.389 9.391 9.401 9.405 9.423 9.433 9.445 9.449 9.469 9.475 9.479 9.483 9.493 9.499 9.537 9.525 9.518 9.519 9.503 9.482 9.476 9.469 9.461 9.462

1-2 2 9.456 9.447 9.440 9.437 9.446 9.459 9.474 9.490 9.504 9.518 9.531 9.541 9.549 9.571 9.581 9.597 9.623 9.657 9.685 9.718 9.749 9.775 9.796 9.813 9.577

1-3 3 9.829 9.841 9.849 9.861 9.880 9.899 9.907 9.923 9.939 9.949 9.967 9.977 9.981 9.977 9.981 9.982 9.985 9.983 9.981 9.975 9.987 9.984 9.980 9.974 9.941

1-4 4 9.961 9.943 9.933 9.928 9.914 9.919 9.908 9.903 9.893 9.885 9.875 9.873 9.852 9.842 9.809 9.791 9.794 9.789 9.782 9.781 9.769 9.787 9.815 9.823 9.857

1-5 5 9.839 9.855 9.865 9.882 9.891 9.901 9.905 9.907 9.925 9.932 9.939 9.951 9.961 9.967 9.973 9.978 9.991 0.003 0.021 0.036 0.061 0.077 0.100 0.113 9.961

Figura 4.4: Resultado de la extracción de líneas con redes neuronales visto en Transkribus. En la parte inferior presentamos parte del *ground truth* de la imagen.

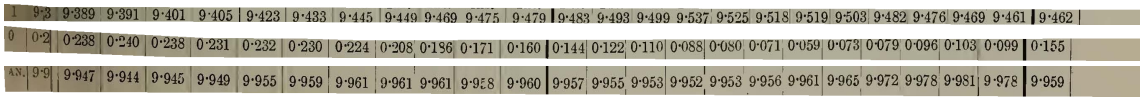


Figura 4.5: Líneas 1,30 y 31 de la tabla del mes de Junio de 1901. Nótese como en la línea 30 la información de la parte izquierda de la tabla aparece incompleta por la deformación de la hoja.

4.3 Descripción del modelo óptico

Teniendo las tablas seccionadas por líneas, el siguiente paso es el de reconocer los carácteres que hay en cada imagen para hacer la transcripción. Para la consecución de este objetivo, utilizamos una red neuronal artificial con la arquitectura que describimos en esta sección.

4.3.1. Capas de convolución

Las primeras cuatro capas de nuestro modelo son convolucionales, con función de activación *LeakyReLU*. Estas capas tienen como objetivo extraer características de la imagen, sirviendo como herramienta de preprocesamiento de la imagen guiado por los datos. Para conseguirlo, por cada región de la imagen, la capa realiza diversas operaciones matemáticas para producir un único valor numérico en el mapa de características de salida. Esta operación matemática, realizada sobre regiones de tamaño definido, se denomina filtro convolucional. Para ilustrar mejor el funcionamiento de una capa de convolución, proponemos la representación de la figura 4.6.

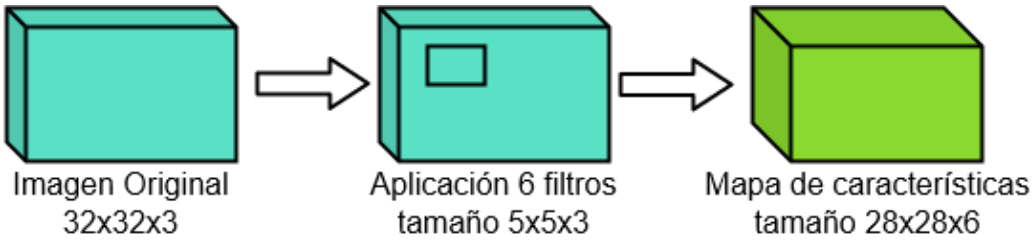


Figura 4.6: Representación de una capa de convolución donde se aplican 6 filtros. Nótese como la profundidad del mapa de características coincide con el número de filtros aplicado.

La dimensión de las capas va en aumento, de tal manera que las primeras capas se centran en los detalles más pequeños de la imagen, como partes de los trazos de cada número. Las capas posteriores, que tienen como entrada los mapas de características producidos por las capas previas, analizan partes mayores de la imagen, intentando extraer las formas características de los caracteres.

Tras cada capa de convolución suele haber una capa *pooling*, que reduce la dimensionalidad de los datos agrupándolos según un tamaño y criterio a especificar. El criterio utilizado en nuestro sistema es *max pooling*, lo que significa que de cada región en el mapa de características se escoge el valor numérico más alto. Cabe comentar que, pese al impacto negativo que pudiera tener la compresión de datos, el uso de estas capas aumenta la velocidad de procesamiento del sistema por lo que su uso es aconsejado. A fin de ilustrar mejor el funcionamiento de una capa *pooling* por maximización, proponemos la figura 4.7.

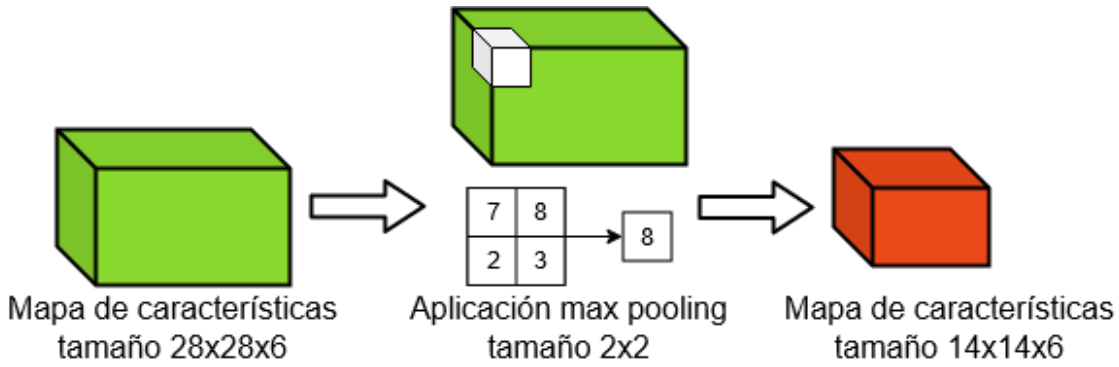


Figura 4.7: Representación de una capa *pooling* por maximización de tamaño 2x2. Nótese como la profundidad del mapa de características permanece constante.

4.3.2. Capas LSTM

Antes de estar en posición de explicar correctamente lo que es una red *Long Short Term Memory* (LSTM) y, consecuentemente, las capas LSTM, debemos conocer qué es una red neuronal artificial recurrente. Las redes recurrentes[2] vienen motivadas por la necesidad de recordar información previa, algo especialmente útil cuando se está haciendo reconocimiento de texto en imagen o tareas de procesamiento del lenguaje natural. Para conseguir este objetivo, las redes recurrentes presentan ciclos de tal forma que la salida de la red sirve como entrada para la computación inmediatamente posterior.

A fin de lograr una mejor visualización de la arquitectura de una red recurrente pronemos la representación simplificada de la figura 4.8, donde x_t es la muestra escogida en el instante t, y_t la salida de la red en dicho instante y a_t un estado intermedio, a partir del cual se calcula la salida, que sirve como entrada para la computación inmediatamente posterior.

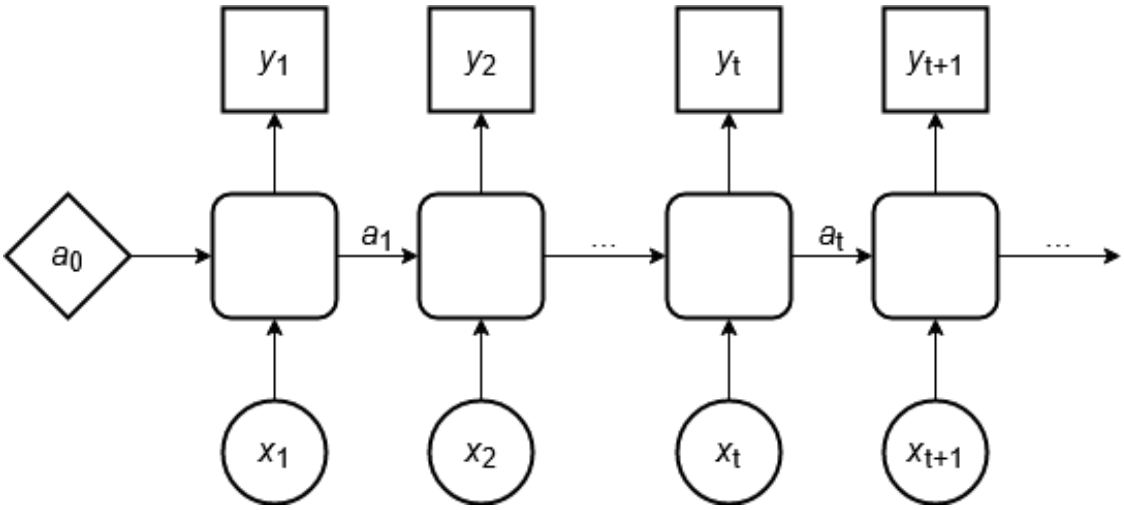


Figura 4.8: Representación simplificada de una red recurrente.

Las redes LSTM surgen como una mejora de las redes recurrentes clásicas al incoporar mecanismos para lidiar con el problema del desvanecimiento del gradiente. Este problema, descrito de manera sucinta, surge al trabajar con grandes volúmenes de datos y produce que el contenido de la memoria se pierda si se encuentra muy alejado del punto de procesamiento actual. En nuestro modelo utilizamos tres capas de tipo LSTM a continuación de las cuatro capas convolucionales para incluir el dominio temporal como información a considerar en el modelo óptico. Dichas capas LSTM tienen como función de activación la *Softmax*, lo que nos permite modelar una distribución de probabilidad para cada carácter.

4.3.3. Entrenamiento de la red y cálculo de probabilidades

La red recibe como entrada una secuencia de vectores por cada línea de la tabla a reconocer. Estos vectores son partes de la imagen, resultado de hacer una división por columnas de manera periódica de izquierda a derecha de la línea, que permiten simular la información temporal. Como se puede imaginar, habrán carácteres que ocupen más de una división.

[2]Explicación detallada de redes recurrentes y LSTM: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Para trabajar correctamente con estos datos, hacemos uso de la *Connectionist Temporal Classification* (CTC) [5]. Esta técnica nos permite efectuar el cálculo de la función de pérdida asociada a la transcripción de un carácter y los vectores que ocupa, valor que utilizamos para entrenar la red con el algoritmo de retropropagación.

No obstante, este no es el único uso que se le da a la CTC. Al calcular la transcripción más probable haciendo una exploración con el algoritmo de búsqueda en haz, los cálculos de las probabilidades se hacen teniendo en cuenta todos los posibles alineamientos para una transcripción. Por ello, podemos decir que utilizamos la CTC tanto para el entrenamiento de la red como para el cálculo de la probabilidad a la hora de la transcripción.

4.4 Decodificación con SFSA

Una técnica muy estandarizada para mejorar los resultados de un sistema HTR es el uso de un *Stochastic Finite State Automata* (SFSA) que permite modelar información contextual durante la decodificación de la señal. Esta técnica está importada del reconocimiento del habla, y ha demostrado ser de gran utilidad en sistemas donde la regularidad de los datos es algo escasa. Este SFSA aglutina cuatro niveles de conocimiento: morfológico, léxico, sintáctico y semántico. Sin embargo, para nuestro sistema solo utilizamos los tres primeros como explicamos a continuación.

4.4.1. Conocimiento morfológico: modelo oculto de Márkov

El primer nivel de conocimiento es el morfológico. Con este nivel se describe los trazos y las estructuras que componen cada carácter a reconocer. La implementación de este conocimiento consiste en un modelo oculto de Márkov por cada carácter que se entrena sobre un conjunto de muestras a fin de aprender como se escribe cada carácter.

En nuestro caso dicho conjunto de muestras es el mismo que el que se usa para entrenar el modelo óptico. El entrenamiento de las probabilidades de emisión se realiza utilizando el algoritmo CTC. La topología de los modelos de Márkov es estricatamente lineal, sin saltos; como se ilustra en la figura 4.9.

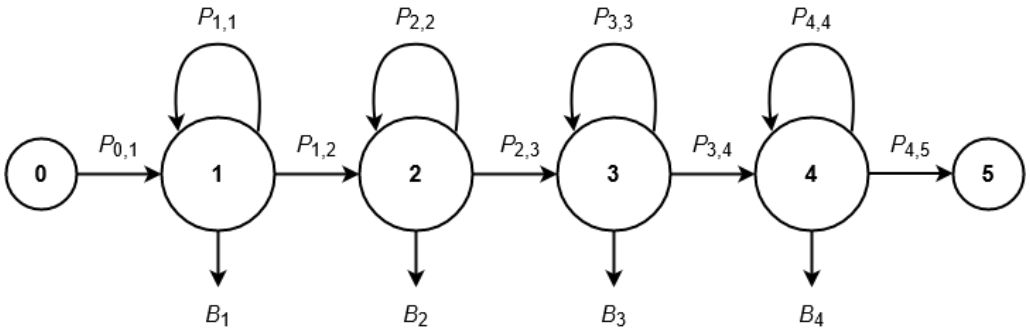


Figura 4.9: Prototipo de modelo oculto de Márkov que implementa el nivel de conocimiento morfológico.

Dicho modelo podría, por ejemplo, describir la manera en la que se escribe el carácter '9'. Las probabilidades de transición $P_{x,y}$, desde un punto de vista teórico, se deberían entrenar. No obstante, hemos decidido dejar dichas probabilidades fijas, al dar buenos resultados en la práctica, y entrenar únicamente las probabilidades de emisión B_x de cada estado x.

4.4.2. Conocimiento léxico: autómata finito de estados

El segundo nivel de conocimiento es el léxico. En este nivel se especifica las secuencias de carácteres que pueden servir para escribir cada palabra del vocabulario. En nuestro caso la construcción del modelo de lenguaje se hace de tal manera que cada carácter es considerado como una palabra completa. Por tanto, como es evidente, cada palabra del vocabulario admite una única forma léxica.

El motivo para escoger esta arquitectura, en lugar de utilizar reconocimiento aislado, es sencillo: la cantidad de muestras es insuficiente para entrenar un sistema con el vocabulario tan extenso que se generaría al considerar las palabras completas, por lo que el modelo de lenguaje sería incapaz de generalizar correctamente.

La implementación de este nivel suele consistir en un SFSA donde cada transición, asociada con un carácter, está representada por el modelo oculto de Márkov correspondiente aprendido en el nivel morfológico. En nuestro caso, al existir una única transcripción para cada palabra, el SFSA pierde su componente estocástica. En la figura 4.10 se puede observar el autómata finito de estados asociado con el dígito '9'.

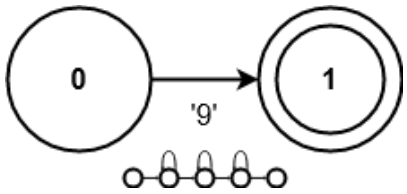


Figura 4.10: Representación del autómata finito de estados para la palabra '9'. Nótese como en la transición figura el modelo oculto de Márkov correspondiente al carácter.

4.4.3. Conocimiento sintáctico: n-grama

El tercer nivel de conocimiento es el sintáctico. Este nivel describe la manera en la que se han de concatenar palabras para la correcta formación de frases. En nuestro caso modelamos dicha información con un n-grama. Dicho n-grama ve su representación como un SFSA con tantos estados como combinaciones de $n-1$ palabras, en nuestro caso caracteres, sea posible producir a partir del vocabulario.

El caso más sencillo de explicar es el bigrama, donde hay tantos estados como palabras. La probabilidad de aparición de la siguiente palabra viene influenciada por la palabra anterior. En términos específicos, la probabilidad de transición del estado actual al siguiente estado, que identifica una palabra, es la probabilidad de generar dicha palabra dado el contexto. A fin de ilustrar esta relación, exponemos la siguiente figura:

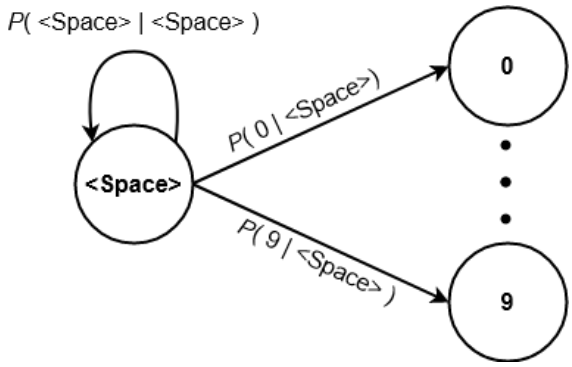


Figura 4.11: Representación de la probabilidad de generación de la palabra posterior a '<Space>'.

A partir de aquí resulta sencillo generalizar el modelo. En un n-grama cada estado representa una secuencia de $n-1$ palabras. La probabilidad de generar la próxima palabra viene determinada por las $n-1$ palabras que la preceden, ya que se transita desde el estado que contiene el contexto a aquel que incluye la siguiente palabra.

Como se puede intuir, el nivel de conocimiento sintáctico aglutina los dos niveles anteriores. Las transiciones de un estado a otro en el n-grama vienen representadas como los FSA propios del conocimiento léxico, que a su vez incluyen los modelos ocultos de Márkov en sus transiciones a nivel de carácter. De esta manera, nuestro modelo de lenguaje puede entenderse como un gran SFSA resultante de la composición de los tres niveles de conocimiento que hemos descrito con anterioridad.

A la hora del reconocimiento el modelo de lenguaje aporta una probabilidad a priori independientemente de la muestra con la que se trabaje. Al mismo tiempo, el modelo óptico calcula la probabilidad de cada hipótesis partiendo de una secuencia de vectores de características. Sobre dicha secuencia de vectores se ha de aplicar el algoritmo de búsqueda en haz para seleccionar en cada instante de tiempo la hipótesis que tenga mayor probabilidad de ser generada.

4.5 Ajuste del GSF y de la WIP

Al utilizar un modelo de lenguaje, tenemos que hacer frente a dos problemas. El primero de todos es que la aportación del modelo de lenguaje a la estimación de la probabilidad ha de estar en la misma escala que la aportación del modelo óptico. Para ello, se entrena el GSF. El segundo problema es que el sistema tiene preferencia por las hipótesis más cortas al hacer el cálculo de probabilidades a posteriori. Para atenuar este sesgo, tenemos que ajustar la WIP sobre un conjunto de validación tal y como se hace con el GSF.

Ambos parámetros se estiman mediante el uso del método de optimización Simplex, de la librería SciPy,[3] sobre un reducido conjunto de validación. Dicho conjunto de validación se reutiliza para hacer unas pocas iteraciones extra sobre el modelo óptico, aprovechando los datos al máximo.

[3]Página web de SciPy: https://www.scipy.org/

CAPÍTULO 5
Experimentación y resultados

En este capítulo centramos nuestra atención en comentar aspectos más específicos de los datos a utilizar, la arquitectura de los modelos a evaluar y el proceso y resultados de la experimentación.

5.1 Descripción del conjunto de datos

El corpus en el que se basa nuestra investigación, como ya hemos visto, consiste en datos de presión atmosférica organizados en tablas de números escritos a máquina. Los datos muestran la evolución de la presión atmosférica, habiendo una columna por cada hora del día, en el observatorio de Fort William, Escocia. Cada tabla escaneada corresponde a la evolución en un mes del año, incluyendo las medias ariméticas.

El tamaño del corpus es bastante limitado, al disponer únicamente de datos desde enero de 1898 hasta septiembre de 1904. Concretamente, disponemos de un total de 81 tablas con su transcripción; lo que se traduce en 2545 líneas. El número total de palabras de muestra es, por tanto, 66170.

Si se analiza un poco más la colección de datos, vemos que el conjunto de carácteres a reconocer es bastante reducido: dígitos del 0 al 9, carácteres de la palabra *Mean*, el separador decimal y el espacio en blanco. A este conjunto de caracteres se le añade la cadena vacía para facilitar la implementación. La tabla 5.1 condensa toda esta información sobre el conjunto de datos.

Carácteres diferentes	Núm. de tablas	Núm. de líneas	Núm. de palabras
17	81	2545	66170

Tabla 5.1: Características del conjunto de datos de Fort William.

Otro de los aspectos a destacar de este conjunto, es la estructura de las palabras a reconocer; principalmente números. Los datos de la tabla vienen en formato D·DDD, independientemente de la presión a representar, ya que se elimina el primer dígito de cada número. El resto de palabras a reconocer son los índices de las filas y columnas de cada tabla.

Estos datos, así como su transcripción a formato CSV, son cortesía del galardonado proyecto de voluntariado Old Weather [1]. Este proyecto, que incorpora a multitud de investigadores de diversas instituciones académicas, tiene como objetivo la digitalización de la información meteorológica antigua para su posterior estudio.

5.2 Descripción de los modelos a evaluar

Durante la experimentación se va a evaluar la aportación del modelo de lenguaje y el efecto que tiene aumentar la probabilidad de hacer un bucle en el modelo oculto de Márkov, implementación del nivel de conocimiento morfológico. Además, se va a valorar la importancia del retoque manual de los resultados de la extracción de líneas.

Para conseguir dichos objetivos, entrenamos los modelos con dos conjuntos de datos distintos: uno basado en las líneas extraídas sin corrección manual alguna y el otro con el retoque manual. Dichos modelos pueden incorporar un modelo de lenguaje o no. En el caso de no hacerlo, la transcripción se hace únicamente teniendo en cuenta la aportación del modelo óptico. En el caso de incorporar un modelo de lenguaje, se varía el tamaño del n-grama para tratar de encontrar la n óptima. Además, en los casos en los que se incorpora el modelo de lenguaje, probamos a variar ligeramente la probabilidad de transición de un estado a si mismo dentro del modelo de Márkov. Para ilustrar con mayor claridad el conjunto de modelos que se va a evaluar, proponemos la tabla 5.2.

Conjunto de datos	Modelo de lenguaje	Probabilidad de bucle
Lineas retocadas	Sin LM	-
	n-grama $n \in \{1..15\}$	0,5
		0,6
		0,7
Lineas sin retocar	Sin LM	-
	n-grama $n \in \{1..15\}$	0,5
		0,6
		0,7

Tabla 5.2: Desglose de los diferentes modelos evaluados.

Cabe comentar que, independientemente de la configuración escogida, el aprendizaje de los parámetros del modelo óptico permanece inalterado. Se sigue el mismo modelo expuesto en el capítulo previo: cuatro capas convolucionales que preceden a tres capas LSTM. La única variabilidad que se observará a este respecto será la proveniente del conjunto de datos.

5.3 Proceso experimental

El proceso que seguimos a la hora de la obtención de resultados es validación cruzada en nueve bloques. La división entre conjunto de entrenamiento y test se va a hacer partiendo de las tablas como unidad y no a partir de las líneas extraídas. Al disponer de 81 imágenes el único divisor válido es nueve.

En cada partición un total de 72 meses se destinarán a entrenamiento, mientras que los nueve meses restantes se destinarán a la evaluación del sistema. De los datos destinados a entrenamiento se extraen 250 líneas, independientemente del tamaño del conjunto, para la validación durante el entrenamiento y el ajuste de otros parámetros del sistema. Este reparto produce la distribución que se ilustra en la tabla 5.3.

Número de líneas	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9
Entrenamiento	2013	2013	2012	2011	2013	2013	2012	2011	2012
Validación	250	250	250	250	250	250	250	250	250
Evaluación	282	282	283	284	282	282	283	284	283

Tabla 5.3: Distribución de las líneas en cada partición generada.

Cabe destacar que a la hora de la distribución de las líneas no se ha hecho un proceso de aleatorización tal y como se haría con prácticamente cualquier otro corpus. Esto es porque la homogeneidad de los datos hace innecesario disponer de muestras de todas las imágenes a la hora del entrenamiento. Al mantener la continuidad en los datos, es posible incluir más restricciones en el modelo para obtener mejores resultados.

A la hora de evaluar el rendimiento del sistema hacemos el cálculo del CER y del WER. Ya que se está utilizando validación cruzada, dicho cálculo se tiene que hacer sobre cada instanciación que se entrena. El rendimiento del sistema se computa como la media de las nueve medidas.

5.4 Resultados

Los resultados que se exponen corresponden a la evaluación de las arquitecturas planteadas en términos del CER y del WER en formato porcentaje. En las figuras 5.1 y 5.2 se puede observar la evolución del CER y del WER al entrenar y evaluar el sistema con el conjunto de líneas con retoque manual. Las figuras 5.3 y 5.4 muestran dicha evolución al utilizar el conjunto de datos sin retoque manual. No obstante, incluimos dos tablas al final de este documento con todos los resultados que se pueden observar en las gráficas.

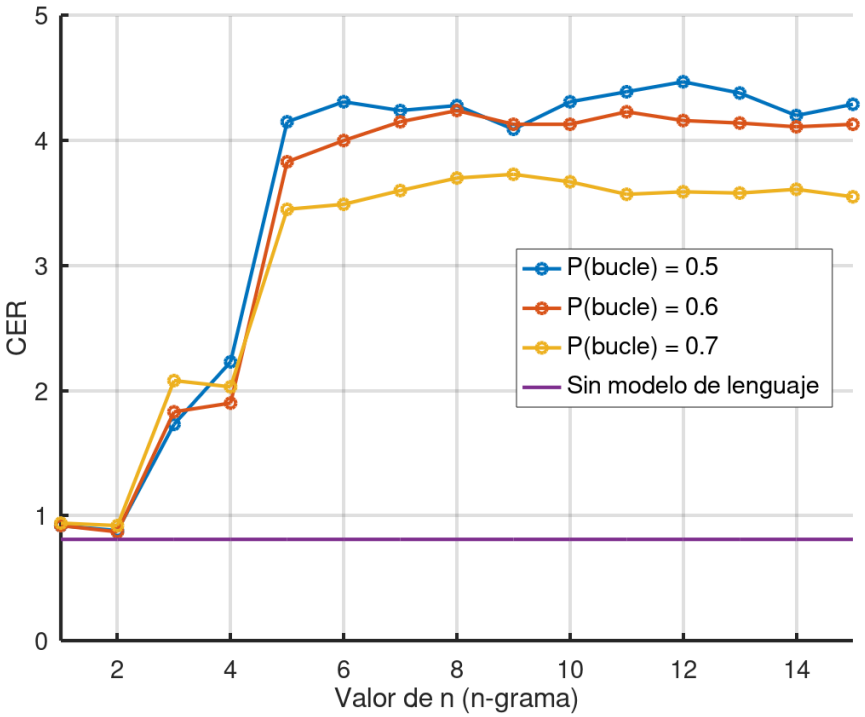


Figura 5.1: Evolución del CER, entrenando con las líneas retocadas, en función del valor de n en el n-grama.

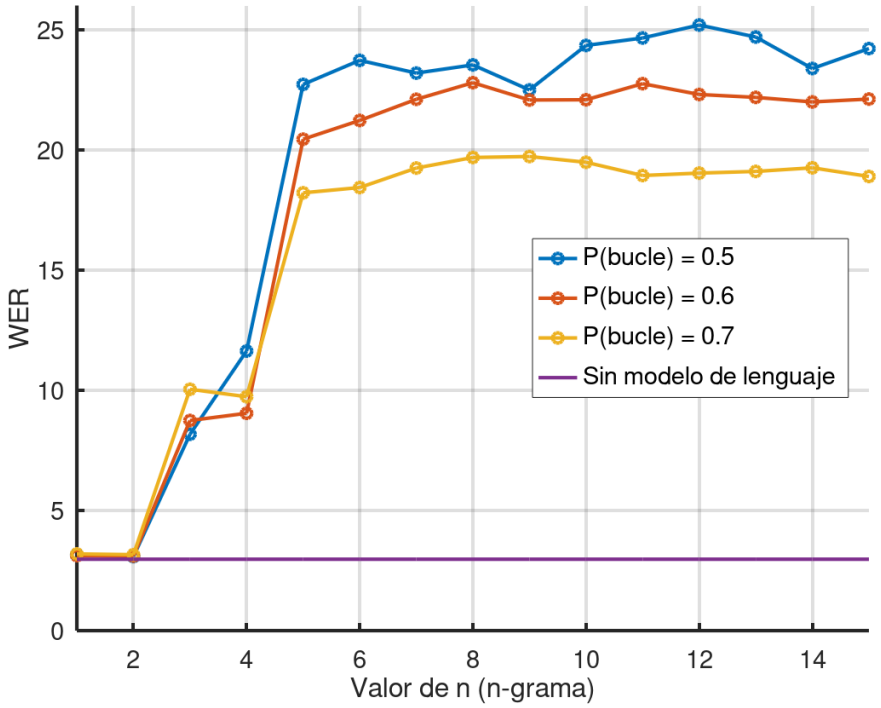


Figura 5.2: Evolución del WER, entrenando con las líneas retocadas, en función del valor de n en el n-grama.

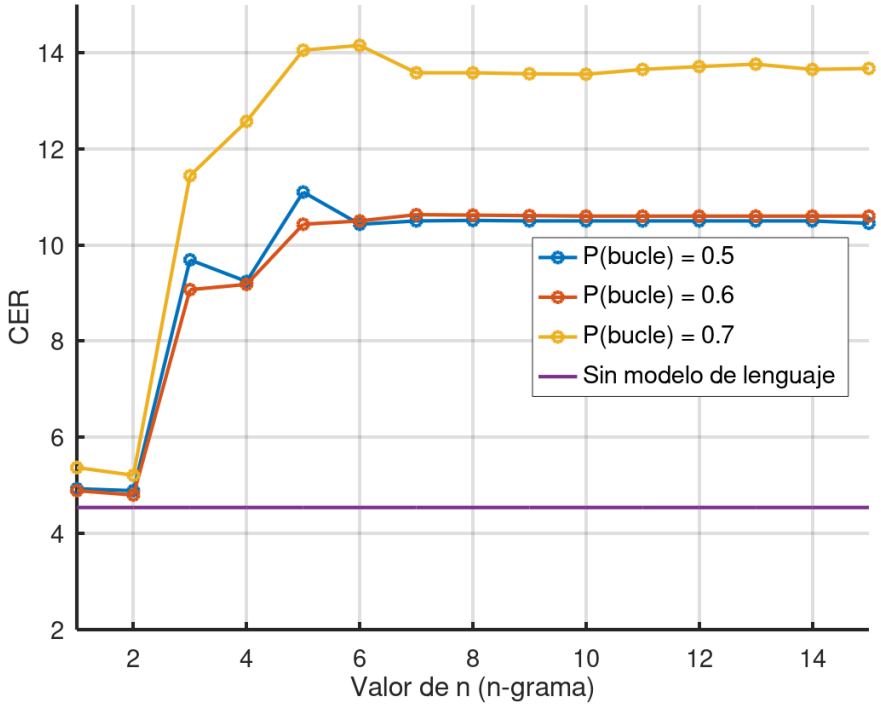


Figura 5.3: Evolución del CER, entrenando con las líneas sin retoque, en función del valor de n en el n-grama.

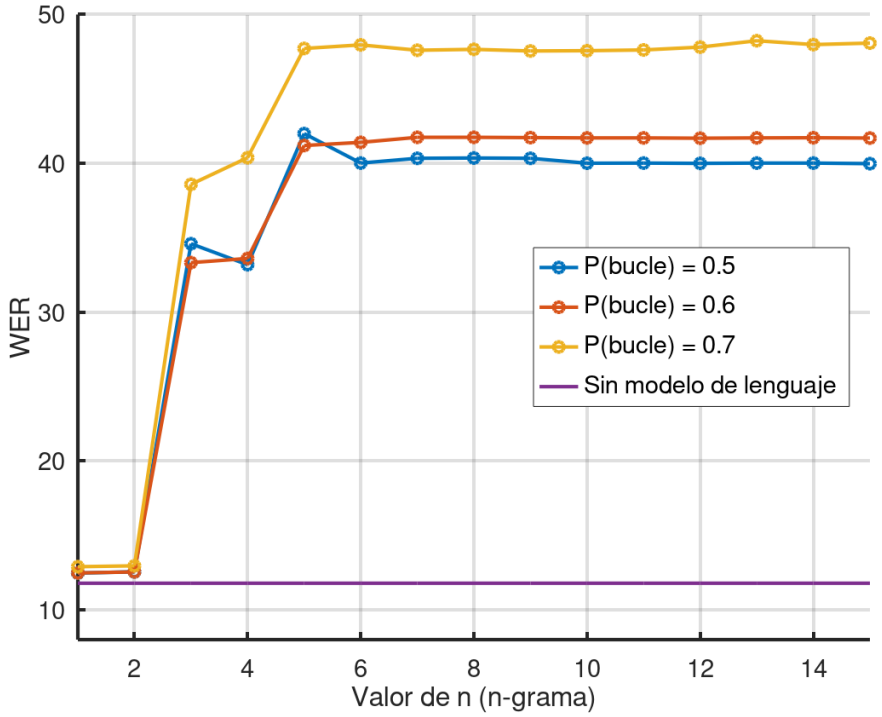


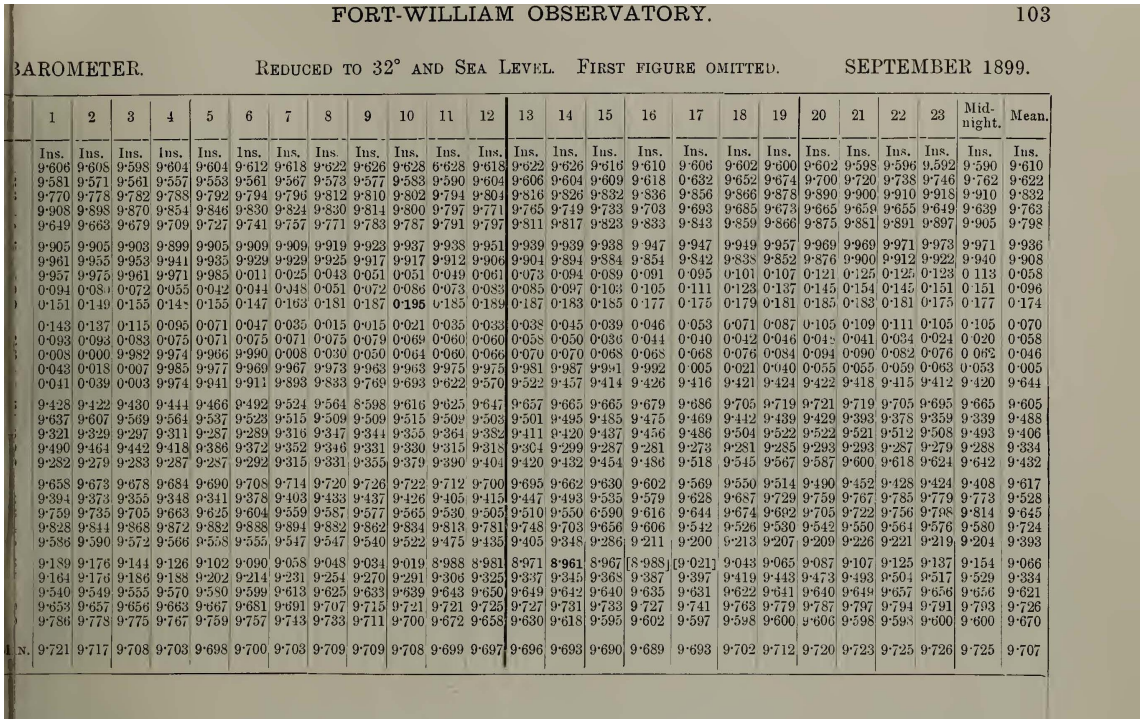
Figura 5.4: Evolución del WER, entrenando con las líneas sin retoque, en función del valor de n en el n-grama.

5.5 Análisis de los resultados

A partir de los resultados obtenidos, así como al realizar un análisis de la salida del sistema, podemos comprobar algunas de las hipótesis iniciales que habíamos planteado. Cabe comentar que durante el análisis de los resultados nos centraremos principalmente en el índice de error a nivel de palabra, ya que al trabajar con información de tipo numérico un error a nivel de carácter implica una transcripción incorrecta; mientras que cuando se trabaja con texto hay palabras con errores que se pueden entender por el contexto.

Una de las primeras observaciones que podemos realizar es como aquellos sistemas que no utilizan ningún modelo de lenguaje resultan más eficaces para esta tarea independientemente del conjunto de datos utilizado. Esto se debe principalmente a, como se ha comentado previamente, la destacable regularidad de los datos. Al contar con formas idénticas para cada carácter, el modelo óptico es capaz de identificar los diferentes símbolos que aparecen en cada imagen, a excepción de aquellos que no aparecen en la imagen por fallos en el escaneado y los que se encuentran en zonas muy ruidosas. Un ejemplo particular de este tipo de muestra lo encontramos en la figura 5.5, donde las palabras que deberían aparecer en el margen izquierdo de la tabla no han sido escaneadas y aparecen corchetes en mitad de la tabla.

El hecho de añadir un modelo de lenguaje hace que se tenga en cuenta la probabilidad a priori de generación de una secuencia de palabras, en vez de basarnos únicamente en la probabilidad condicional que aporta el modelo óptico. En estos casos donde el modelo óptico es capaz de generalizar sin mayor complicación, dicha probabilidad a priori no es lo suficientemente útil e incluso puede entenderse como ruido en algunos casos.



FORT-WILLIAM OBSERVATORY. 103

BAROMETER. REDUCED TO 32° AND SEA LEVEL. FIRST FIGURE OMITTED. SEPTEMBER 1899.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Mid-night.	Mean.
	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.	Ins.
	9·606	9·608	9·598	9·604	9·604	9·612	9·618	9·622	9·626	9·628	6·628	9·618	9·622	9·626	9·616	9·610	9·606	9·602	9·600	9·602	9·598	9·596	9.592	9·590	9·610
	9·581	9·571	9·561	9·557	9·553	9·561	9·567	9·573	9·577	9·583	9·590	9·604	9·606	9·604	9·609	9·618	0·632	9·652	9·674	9·700	9·720	9·738	9·746	9·762	9·622
	9·770	9·778	9·782	9·788	9·792	9·794	9·796	9·812	9·810	9·802	9·794	9·804	9·816	9·826	9·832	9·836	9·856	9·866	9·878	9·890	9·900	9·910	9·918	9·910	9·832
	9·908	9·898	9·870	9·854	9·846	9·830	9·824	9·830	9·814	9·800	9·797	9·771	9·765	9·749	9·733	9·703	9·693	9·685	9·673	9·665	9·659	9·655	9·649	9·639	9·763
	9·649	9·663	9·679	9·709	9·727	9·741	9·757	9·771	9·783	9·787	9·791	9·797	9·811	9·817	9·823	9·833	9·843	9·859	9·866	9·875	9·881	9·891	9·897	9·905	9·798
	9·905	9·905	9·903	9·899	9·905	9·909	9·909	9·919	9·923	9·937	9·938	9·951	9·939	9·939	9·938	9·947	9·947	9·949	9·957	9·969	9·969	9·971	9·973	9·971	9·936
	9·961	9·955	9·953	9·941	9·935	9·929	9·929	9·925	9·917	9·917	9·912	9·906	9·904	9·894	9·884	9·854	9·842	9·838	9·852	9·876	9·900	9·912	9·922	9·940	9·908
	9·957	9·975	9·961	9·971	9·985	0·011	0·025	0·043	0·051	0·051	0·049	0·061	0·073	0·094	0·089	0·091	0·095	0·101	0·107	0·121	0·125	0·125	0·123	0 113	0·058
	0·094	0·084	0·072	0·055	0·042	0·044	0·048	0·051	0·072	0·086	0·073	0·083	0·085	0·097	0·103	0·105	0·111	0·123	0·137	0·145	0·154	0·145	0·151	0·151	0·096
	0·151	0·149	0·155	0·145	0·155	0·147	0·163	0·181	0·187	0·195	0·185	0·189	0·187	0·183	0·185	0·177	0·175	0·179	0·181	0·185	0·183	0·181	0·175	0·177	0·174
	0·143	0·137	0·115	0·095	0·071	0·047	0·035	0·015	0·015	0·021	0·035	0·033	0·038	0·045	0·039	0·046	0·053	0·071	0·087	0·105	0·109	0·111	0·105	0·105	0·070
	0·093	0·093	0·083	0·075	0·071	0·075	0·071	0·075	0·079	0·069	0·060	0·060	0·058	0·050	0·036	0·044	0·040	0·042	0·046	0·045	0·041	0·034	0·024	0 020	0·058
	0·008	0·000	9·982	9·974	9·966	9·990	0·008	0·030	0·050	0·064	0·060	0·066	0·070	0·070	0·068	0·068	0·068	0·076	0·084	0·094	0·090	0·082	0·076	0 062	0·046
	0·043	0·018	0·007	9·985	9·977	9·969	9·967	9·973	9·963	9·963	9·975	9·975	9·981	9·987	9·991	9·992	0·005	0·021	0·040	0·055	0·055	0·059	0·063	0·053	0·005
	0·041	0·039	0·003	9·974	9·941	9·911	9·893	9·883	9·769	9·693	9·622	9·570	9·522	9·457	9·414	9·426	9·416	9·421	9·424	9·422	9·418	9·415	9·412	9·420	9·644
	9·428	9·422	9·430	9·444	9·466	9·492	9·524	9·564	8·598	9·616	9·625	9·647	9·657	9·665	9·665	9·679	9·686	9·705	9·719	9·721	9·719	9·705	9·695	9·665	9·605
	9·637	9·607	9·569	9·564	9·537	9·523	9·515	9·509	9·509	9·515	9·509	9·503	9·501	9·495	9·485	9·475	9·469	9·442	9·439	9·429	9·393	9·378	9·359	9·339	9·488
	9·321	9·329	9·297	9·311	9·287	9·289	9·316	9·347	9·344	9·353	9·364	9·382	9·411	9·420	9·437	9·456	9·486	9·504	9·522	9·522	9·521	9·512	9·508	9·493	9·406
	9·490	9·464	9·442	9·418	9·386	9·372	9·352	9·346	9·331	9·330	9·315	9·318	9·304	9·299	9·287	9·281	9·273	9·281	9·285	9·293	9·293	9·287	9·279	9·288	9·334
	9·282	9·279	9·283	9·287	9·287	9·292	9·315	9·331	9·355	9·379	9·390	9·404	9·420	9·432	9·454	9·486	9·518	9·545	9·567	9·587	9·600	9·618	9·624	9·642	9·432
	9·658	9·673	9·678	9·684	9·690	9·708	9·714	9·720	9·726	9·722	9·712	9·700	9·695	9·662	9·630	9·602	9·569	9·550	9·514	9·490	9·452	9·428	9·424	9·408	9·617
	9·394	9·373	9·355	9·348	9·341	9·378	9·403	9·433	9·437	9·426	9·405	9·415	9·447	9·493	9·535	9·579	9·628	9·687	9·729	9·759	9·767	9·785	9·779	9·773	9·528
	9·759	9·735	9·705	9·663	9·625	9·604	9·559	9·587	9·577	9·565	9·530	9·505	9·510	9·550	6·590	9·616	9·644	9·674	9·692	9·705	9·722	9·756	9·798	9·814	9·645
	9·828	9·844	9·868	9·872	9·882	9·888	9·894	9·882	9·862	9·834	9·813	9·781	9·748	9·703	9·656	9·606	9·542	9·526	9·530	9·542	9·550	9·564	9·576	9·580	9·724
	9·586	9·590	9·572	9·566	9·558	9·555	9·547	9·547	9·540	9·522	9·475	9·435	9·405	9·348	9·286	9·211	9·200	9·213	9·207	9·209	9·226	9·221	9·219	9·204	9·393
	9·189	9·176	9·144	9·126	9·102	9·090	9·058	9·048	9·034	9·019	8·988	8·981	8·971	8·961	8·967	[8·988]	[9·021]	9·043	9·065	9·087	9·107	9·125	9·137	9·154	9·066
	9·164	9·176	9·186	9·188	9·202	9·214	9·231	9·254	9·270	9·291	9·306	9·325	9·337	9·345	9·368	9·387	9·397	9·419	9·443	9·473	9·493	9·504	9·517	9·529	9·334
	9·540	9·549	9·555	9·570	9·580	9·599	9·613	9·625	9·633	9·639	9·643	9·650	9·649	9·642	9·640	9·635	9·631	9·622	9·641	9·640	9·649	9·657	9·656	9·656	9·621
	9·653	9·657	9·656	9·663	9·667	9·681	9·691	9·707	9·715	9·721	9·721	9·725	9·727	9·731	9·733	9·727	9·741	9·763	9·779	9·787	9·797	9·794	9·791	9·793	9·726
	9·786	9·778	9·775	9·767	9·759	9·757	9·743	9·733	9·711	9·700	9·672	9·658	9·630	9·618	9·595	9·602	9·597	9·598	9·600	9·606	9·598	9·598	9·600	9·600	9·670
M.	9·721	9·717	9·708	9·703	9·698	9·700	9·703	9·709	9·709	9·708	9·699	9·697	9·696	9·693	9·690	9·689	9·693	9·702	9·712	9·720	9·723	9·725	9·726	9·725	9·707

Figura 5.5: Datos recogidos durante el mes de septiembre de 1899.

Si nos centramos en los resultados que se obtienen al emplear un unigrama comprobamos que la información que aporta el modelo de lenguaje es incluso perjudicial. Esto se debe a que la probabilidad a priori que estima dicho modelo es la probabilidad de aparición de cada carácter independientemente del contexto. En otras palabras, el unigrama calcula la probabilidad de generación de cada carácter en función del número de veces que ha aparecido previamente.

El bigrama, por otra parte, sí que aporta cierta información de utilidad al sistema. Al modelar la probabilidad a priori de generación de un carácter en función del carácter anterior, el sistema es capaz de aprender que tras cada espacio en blanco puede aparecer un cero, un ocho o un nueve. Otra de las características de los datos que el modelo debería ser capaz de identificar es que, al ser estos dígitos los primeros de cada número, es muy común que después de ellos haya un separador decimal. Sin embargo, el resto de información que podría dar este modelo es casi equivalente a la que aporta el unigrama ya que no hay suficiente contexto para ayudar al modelo óptico. Por ello vemos una mejoría tanto en el WER como en el CER con respecto a los resultados del unigrama, aunque el rendimiento se queda ligeramente por detrás del de aquellos sistemas sin modelo de lenguaje.

Siguiendo el razonamiento expuesto, la primera hipótesis que se puede tener es que conforme aumentamos la complejidad del modelo de lenguaje también aumenta la eficacia del sistema al dar más información contextual. El motivo por el que esta evolución no se vislumbra en nuestro caso es el reducido tamaño del conjunto de datos con el que se está trabajando. Al incrementar la complejidad de los modelos de forma exponencial, resulta imposible aprender todos sus parámetros con tan pocas muestras. Es por ello que, conforme aumentamos el valor de n en el n-grama, aumentan los índices de error muy rápidamente.

Sin embargo, se puede observar como los índices de error tienden a estabilizarse ya que, llegada cierta complejidad, los modelos resultantes tienden a sobreajustarse de igual manera: asignando probabilidades a priori iguales a uno a los casos que se han observado durante el entrenamiento, dejando aquellos que no se han visto sin probabilidad de generación. Este comportamiento se observa en todos los sistemas que utilicen un n-grama como modelo de lenguaje, siempre hay una configuración a partir de la cual el sistema empeora su rendimiento hasta ser incapaz de generalizar. Aunque esto lo combatimos haciendo uso de técnicas de suavizado, el efecto es muy notorio y hace que estos sistemas no sean válidos para la tarea.

Si observamos la evolución de la proporcion entre WER y CER, representada en las figuras 5.6 y 5.7, podemos apreciar como al utilizar un unigrama o un bigrama los errores a nivel de carácter se encuentran ligeramente más concentrados que en un sistema basado únicamente en el uso de un modelo óptico. Conforme aumenta la complejidad del modelo de lenguaje también lo hace la proporción, por lo que los errores a nivel de carácter están más distribuidos.

Al utilizar modelos complejos sobre el conjunto de datos con retoque manual, la proporción del WER asciende hasta ser más del quíntuple del CER. Esto nos indica que por cada palabra incorrecta debe haber únicamente un error de carácter al ser la mayoria de las palabras de longitud igual a cinco. En el caso de las líneas sin retoque, la evolución es similar pero los errores a nivel de carácter se mantienen más concentrados en todo momento.

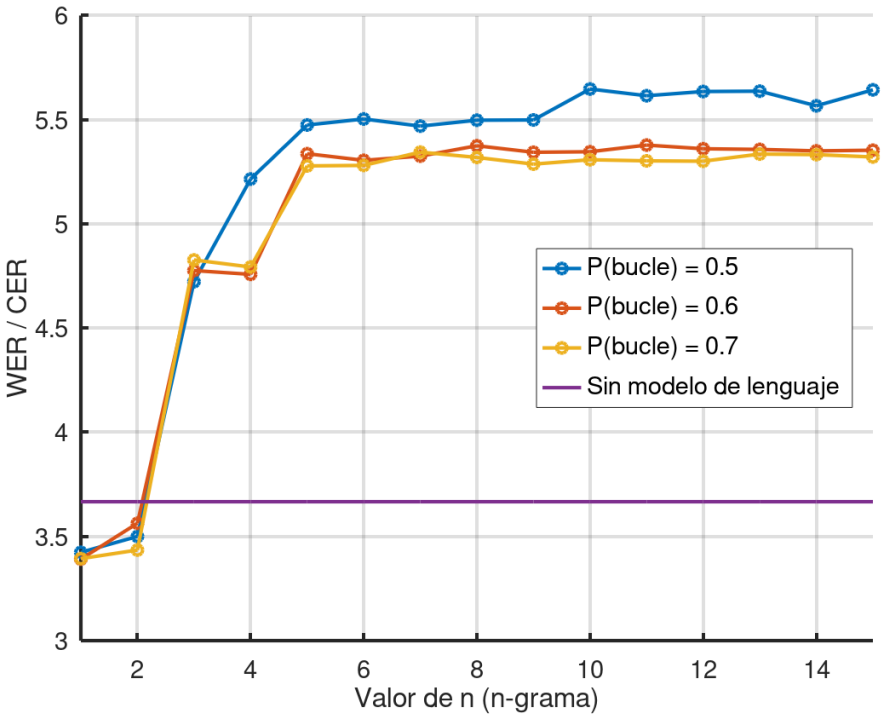


Figura 5.6: Evolución de la proporción entre WER y CER en los modelos entrenados a partir de las líneas con retoque manual.

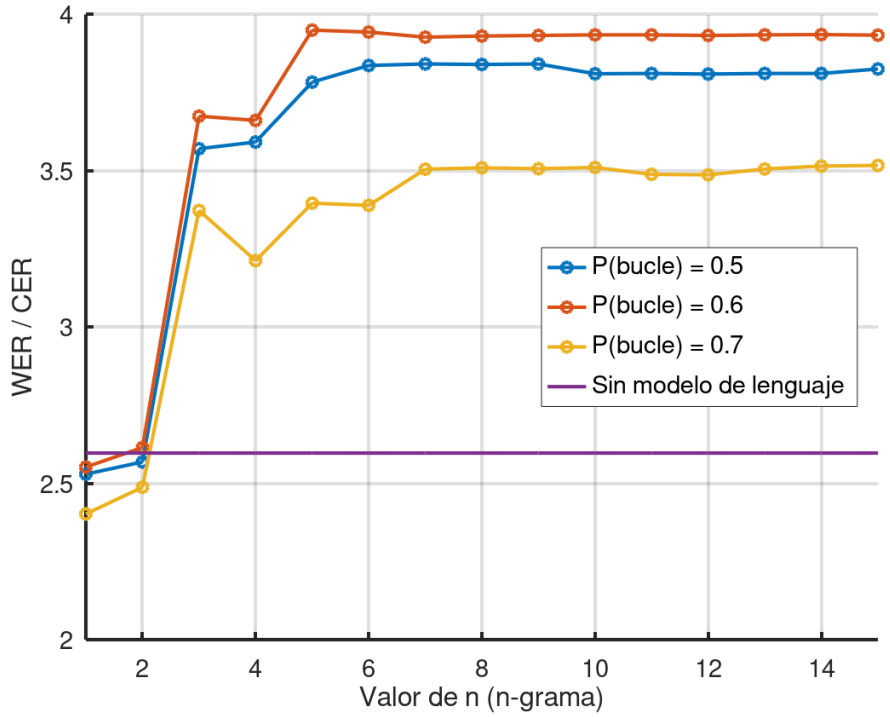


Figura 5.7: Evolución de la proporción entre WER y CER en los modelos entrenados a partir de las líneas sin retoque manual.

Al centrar la atención sobre la salida del sistema y tratando de identificar los errores que comete el mismo, observamos que los sistemas que no incorporan un modelo de lenguaje tienen cierta tendencia a juntar números de columnas distintas. Este caso se da en páginas donde la curvatura hace que la separación entre columnas no se mantenga uniforme, aunque es poco convencional.

Aquellas arquitecturas que incorporan un n-grama complejo, el cual se ha observado que no obtiene resultados aceptables, también sufren este tipo de errores. No obstante, se ha podido comprobar durante la experimentación que el aumento de la probabilidad de bucle en los modelos ocultos de Márkov disminuye la frecuencia de aparición de estos errores.

El error más común en los sistemas que se basan únicamente en el modelo óptico, no obstante, es el de ser incapaz de reconocer los números del margen izquierdo de la tabla. Esto se debe a, como ya hemos visto, una mala digitalización del documento. Otro error que ocurre, aunque con mucha menor frecuencia, es el de confundir números que tengan representaciones parecidas y que, por el paso del tiempo, hayan perdido parte de la tinta. Cabe decir, no obstante, que estos casos son difíciles tanto para un modelo óptico como para el ojo humano.

Los errores que afectan al modelo óptico se incrementan, como es lógico, cuando trabajamos con el conjunto de datos sin retoque manual. En dicho conjunto las líneas originales del sistema de extracción no se modifican, lo que hace que en algunas ocasiones se pierda parte de la información de la tabla. Además, en algunas ocasiones donde la curvatura es más pronunciada, se tiende a incluir partes de los números que hay en las filas adyacentes.

CAPÍTULO 6
Conclusiones

En este capítulo resumimos la información expuesta en cada capítulo del documento, además de hacer una revisión de los objetivos propuestos y ver en qué medida se han cumplido. Por último, proponemos algunas posibles ampliaciones al trabajo realizado.

6.1 Resumen por capítulos

En el primer capítulo hemos expuesto los motivos que nos han impulsado a desarrollar un sistema para una tarea de transcripción automática. Además, hemos expresado el razonamiento seguido a la hora de escoger las redes neuronales artificiales por encima de otros reconocedores.

Seguidamente hemos contextualizado el trabajo, destacando la relación actual que existe entre reconocimiento de texto y reconocimiento del habla. En este capítulo hemos presentado, además, diferentes áreas de aplicación de la transcripción automática y los modelos que se está empleando para la resolución de estas tareas.

A continuación hemos descrito brevemente la teoría sobre la que se fundamenta la solución propuesta, además de definir las métricas CER y WER que se han utilizado para la evaluación del rendimiento del sistema.

En el capítulo posterior hemos detallado la implementación del sistema, comentando las alternativas estudiadas a la hora de procesar los datos. Hemos hecho especial hincapié sobre la arquitéctura del modelo óptico y hemos expuesto las particularidades del modelo de lenguaje utilizado.

Por último, hemos descrito el proceso experimental; además de presentar los resultados obtenidos por las diferentes arquitecturas a evaluar con los dos conjuntos de datos. En este capítulo se ha incluído, además, un análisis exhaustivo del rendimiento del sistema junto con la identificación de los errores más característicos del mismo y sus respectivas causas.

6.2 Objetivos logrados

Consideramos que hemos logrado el objetivo de extraer de manera automática la caja de inclusión de cada fila de las tablas, al obtener resultados aceptables sin el retoque manual de las líneas. No obstante, todavía hay espacio para mejorar al existir una diferencia significativa entre los resultados obtenidos con los datos retocados y los que están sin retocar.

Por otra parte estamos seguros de que hemos logrado entrenar un modelo neuronal capaz de reconocer los carácteres de las imágenes. El rendimiento del sistema, al utilizar las líneas con retoque manual, es aceptable si tenemos en cuenta la naturaleza de los errores. Cabe destacar, una vez más, que hay imágenes donde parte de la información de la tabla no figura.

Por último, resulta evidente que no hemos conseguido aumentar la precisión del sistema utilizando un modelo de lenguaje. Esto se debe a que la cantidad de muestras disponibles es insuficiente para entrenar modelos que aporten información de calidad, además del ya excepcional rendimiento del modelo óptico. No obstante, sigue siendo una técnica de interés para tareas del mismo campo ya que puede dotar de cierta robustez al sistema.

6.3 Trabajo futuro

Una de las alternativas propuestas a la hora de desarrollar la solución fue la de construir una arquitectura que aprovechara la regularidad de los datos. Para ello, diseñamos un SFSA que serviría como aceptor de la gramática que definen los datos.

Si se analiza con detenimiento la estructura de las tablas, se puede observar como existen dos clases de líneas a reconocer. La primera de todas, la más regular, es la línea de cabecera. En esta, se ha de reconocer la secuencia de carácteres '1 2 3...24 Mean'.

El otro prototipo de fila a reconocer es aquel en el que se presentan datos de presión. En cualquier fila de estas características encontramos un número que indica el día del mes al que pertenecen los datos, o bien la palabra 'Mean'. Después, encontramos una secuencia de 25 números de un digito entero y tres decimales.

Aunque en los carácteres inciales el sistema no sabe el camino a seguir, tras reconocer las primeras dos palabras ya se debería seleccionar una de las dos opciones y restringir al componente óptico con la rigidez del autómata. Estas restricciones se manifestarían en forma de probabilidad a priori nula para aquellas hipótesis que no encajen con el esquema impuesto.

No hemos podido implementar esta prometedora técnica porque la herramienta que se pensaba utilizar para el entrenamiento del SFSA gramatical, FSTrain,[1] está desactualizada y no se pudo completar su instalación debido a problemas de compatibilidad, ya que las herramientas de las que depende han continuado actualizándose.

Pese a todo, si se encontrara una alternativa viable para el entrenamiento de autómatas, consideramos que esta estrategia podría reportar buenos resultados al lidiar de manera directa con el problema de la repetición de caracteres en la transcripción.

Otra opción para ampliar el trabajo sería la de, en lugar de fijar las probabilidades de transición en los modelos ocultos de Márkov, ajustarlas mediante el algoritmo Baum-Welch. Esto se haría utilizando la librería OpenGrm Baum-Welch,[2] compatible con el *toolkit* que utilizamos para la construcción de los autómatas finitos de estados: OpenFST.

[1] Repositorio de FSTrain: https://github.com/markusdr/fstrain

[2] Documentación de la librería OpenGrm Baum-Welch: http://www.opengrm.org/twiki/bin/view/GRM/BaumWelchDocs

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología en el proyecto IBEM (TIN2017-91452-EXP) y por la Generalitat Valenciana en el proyecto DeepPattern (PROMETEO/2019/121).

Bibliografía

[1] *Página web de la iniciativa Old Weather*. Disponible en: https://www.oldweather.org/.

[2] Cai, J., L. Peng, Y. Tang, C. Liu y P. Li: *TH-GAN: Generative Adversarial Network Based Transfer Learning for Historical Chinese Character Recognition*. En *2019 International Conference on Document Analysis and Recognition (ICDAR)*, págs. 178–183. IEEE, 2019.

[3] Clausner, C., A. Antonacopoulos, N. Mcgregor y D. Wilson-Nunn: *ICFHR 2018 Competition on Recognition of Historical Arabic Scientific Manuscripts – RASM2018*. En *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, págs. 471–476, 2018.

[4] Furkan Biten, A., R. Tito, A. Mafla, L. Gomez, M. Rusiñol, M. Mathew, C. Jawahar, E. Valveny y D. Karatzas: *ICDAR 2019 Competition on Scene Text Visual Question Answering*. arXiv preprint arXiv:1907.00490, 2019.

[5] Graves, A., S. Fernández, F. Gomez y J. Schmidhuber: *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. En *Proceedings of the 23rd international conference on Machine learning*, págs. 369–376, 2006.

[6] Ingle, R. R., Y. Fujii, T. Deselaers, J. Baccash y A. C. Popat: *A Scalable Handwritten Text Recognition System*. 2019.

[7] Joanna, J. S., K. Sivakumar y N. Sooryadharshini: *Online Handwritten Character Recognition (OHCR) using Deep Learning Convolution Neural Network*.

[8] Jurafsky, D. y J. H. Martin: *Speech and Language Processing (Borrador)*. Disponible en: https://web.stanford.edu/~jurafsky/slp3/, 2020.

[9] Kassis, M. y J. El-Sana: *Learning Free Line Detection in Manuscripts using Distance Transform Graph*. En *2019 International Conference on Document Analysis and Recognition (ICDAR)*, págs. 222–227. IEEE, 2019.

[10] Keret, S., L. Wolf, N. Dershowitz, E. Werner, O. Almogi y D. Wangchuk: *Transductive Learning for Reading Handwritten Tibetan Manuscripts*. En *2019 International Conference on Document Analysis and Recognition (ICDAR)*, págs. 214–221. IEEE, 2019.

[11] Kohavi, R.: *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. págs. 1137–1143. Morgan Kaufmann, 1995.

[12] Kriesel, D.: *A Brief Introduction to Neural Networks*. Disponible en: http://www.dkriesel.com/en/science/neural_networks, 2007.

[13] Mocholí Calvo, C.: *Desarrollo y experimentación de un sistema de aprendizaje profundo para redes neuronales convolucionales y recurrentes*, 2008.

[14] Quirós, L.: *Multi-Task Handwritten Document Layout Analysis*. CoRR, abs/1806.08852, 2018.

[15] Rabiner, L. R. y B. H. Juang: *An Introduction to Hidden Markov Models*. iEEE ASSP MAGAZINE, enero 1986.

[16] Walker, J., Y. Fujii y A. C. Popat: *A web-based ocr service for documents*. En *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria*, vol. 1, 2018.

[17] Watanabe, K., S. Takahashi, Y. Kamaya, M. Yamada, Y. Mekada, J. Hasegawa y S. Miyazaki: *Japanese Character Segmentation for Historical Handwritten Official Documents Using Fully Convolutional Networks*. En *2019 International Conference on Document Analysis and Recognition (ICDAR)*, págs. 934–940. IEEE, 2019.

APÉNDICE A
Tablas de resultados

Durante el capítulo de experimentación y resultados se ha presentado la evolución del sistema en forma de gráfica. No obstante, para observar los resultados reales, proponemos las tablas A.1 y A.2. Se marca los mejores resultados para cada conjunto de datos en negrita, que corresponden con el rendimiento del sistema sin modelo de lenguaje en ambos casos.

Modelo de lenguaje	Probabilidad de bucle					
	0,5		0,6		0,7	
	CER	WER	CER	WER	CER	WER
Sin modelo de lenguaje	**0,81**	**2,97**	**0,81**	**2,97**	**0,81**	**2,97**
1-grama	0,92	3,15	0,92	3,12	0,94	3,19
2-grama	0,88	3,08	0,87	3,10	0,92	3,16
3-grama	1,73	8,17	1,83	8,74	2,08	10,04
4-grama	2,23	11,63	1,90	9,04	2,03	9,73
5-grama	4,15	22,72	3,83	20,44	3,45	18,21
6-grama	4,31	23,72	4,00	21,22	3,49	18,43
7-grama	4,24	23,19	4,15	22,10	3,60	19,24
8-grama	4,28	23,53	4,24	22,79	3,70	19,68
9-grama	4,09	22,49	4,13	22,07	3,73	19,72
10-grama	4,31	24,34	4,13	22,08	3,67	19,48
11-grama	4,39	24,65	4,23	22,75	3,57	18,93
12-grama	4,47	25,19	4,16	22,30	3,59	19,03
13-grama	4,38	24,69	4,14	22,18	3,58	19,10
14-grama	4,20	23,38	4,11	21,99	3,61	19,25
15-grama	4,29	24,21	4,13	22,11	3,55	18,89

Tabla A.1: Resultados de los sistemas con el conjunto de líneas retocadas manualmente.

Modelo de lenguaje	Probabilidad de bucle					
	0,5		0,6		0,7	
	CER	WER	CER	WER	CER	WER
Sin modelo de lenguaje	**4,54**	**11,79**	**4,54**	**11,79**	**4,54**	**11,79**
1-grama	4,93	12,47	4,89	12,48	5,37	12,90
2-grama	4,89	12,56	4,80	12,55	5,21	12,96
3-grama	9,69	34,60	9,07	33,33	11,44	38,59
4-grama	9,24	33,19	9,18	33,61	12,57	40,38
5-grama	11,10	42,00	10,43	41,20	14,05	47,72
6-grama	10,43	40,02	10,50	41,41	14,15	47,96
7-grama	10,50	40,34	10,63	41,75	13,58	47,60
8-grama	10,51	40,36	10,62	41,75	13,58	47,66
9-grama	10,50	40,34	10,61	41,73	13,56	47,55
10-grama	10,50	40,01	10,60	41,71	13,55	47,57
11-grama	10,50	40,02	10,60	41,71	13,65	47,62
12-grama	10,50	40,00	10,60	41,69	13,71	47,81
13-grama	10,50	40,02	10,60	41,71	13,76	48,24
14-grama	10,50	40,02	10,60	41,72	13,65	47,98
15-grama	10,45	39,98	10,60	41,70	13,67	48,08

Tabla A.2: Resultados de los sistemas con el conjunto de líneas sin retoque manual.