Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

Reconocimiento automático de un censo histórico impreso sin recursos lingüísticos

Automatic Recognition of a printed historical census without linguistic resources

Master´s Degree final work

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital (MIARFID)

Author: Dan Anitei

Tutor: Joan Andreu Sánchez

José Miguel Benedí

Course 2020-2021

Resumen

El reconocimiento automático de documentos históricos impresos es actualmente un problema resuelto para muchas colecciones de datos. Sin embargo, los sistemas de reconocimiento automático de documentos históricos impresos aún deben resolver varios obstáculos inherentes al trabajo con documentos antiguos. La degradación del papel o las manchas pueden aumentar la dificultad del correcto reconocimiento de los caracteres. No obstante, dichos problemas se pueden paliar utilizando recursos lingüísticos para entrenar buenos modelos de lenguaje que disminuyan la tasa de error de los caracteres. En cambio, hay muchas colecciones como la que se presenta en este trabajo, compuestas por tablas que contienen principalmente números y nombres propios, para las que no se dispone, ni se necesita, de un modelo lingüístico. En este trabajo se muestra que el reconocimiento automático puede realizarse con éxito para una colección de documentos sin utilizar ningún recurso lingüístico.

Este proyecto cubre la extracción de información y el proceso de OCR dirigido, especialmente diseñados para el reconocimiento automático de un censo español del siglo XIX, registrado en documentos impresos. Muchos de los problemas relacionados con los documentos históricos se resuelven utilizando una combinación de técnicas clásicas de visión por computador y aprendizaje neuronal profundo. Los errores, como los caracteres mal reconocidos, son detectados y corregidos gracias a la información redundante que contiene el censo. Dada la importancia de este censo español para la realización de estudios demográficos, este trabajo da un paso más e introduce un modelo demostrador que facilita la investigación sobre este corpus mediante la indexación de los datos.

Key words: Reconocimiento Óptico de Caracteres, Visión por Computador, Documentos Históricos Impresos, Censo

Abstract

Automatic recognition of typeset historical documents is currently a solved problem for many collections of data. However, systems for automatic recognition of typeset historical documents still need to address several issues inherent to working with this kind of documents. Degradation of the paper or smudges can increase the difficulty of correctly recognizing characters, problems that can be alleviated by using linguistic resources for training good language models which decrease the character error rate. Nonetheless, there are many collections such as the one presented in this paper, composed of tables that contain mainly numbers and proper names, for which a language model is neither available nor useful. This paper illustrates that automatic recognition can be done successfully for a collection of documents without using any linguistic resources.

The paper covers the information extraction and the targeted OCR process, specially designed for the automatic recognition of a Spanish census from the XIX century, registered in printed documents. Many of the problems related to historical documents are overcame by using a combination of classical computer vision techniques and deep learning. Errors, such as miss-recognized characters, are detected and corrected thanks to redundant information that the census contains. Given the importance of this Spanish census for conducting demographic studies, this paper goes a step forward and introduces a demonstrator model to facilitate researching on this corpus by indexing the data.

Key words: Optical Character Recognition, Computer Vision, Historical Printed Documents, Census

Contents

Appendices

List of Figures

List of Tables

CHAPTER 1

Introduction

The cuneiform alphabet that was first developed in the Mesopotamian era (3000 BC), along with Egyptian hieroglyphs, laid the foundation for what we all know today as handwriting. The development of writing allowed cultures to record events, history, laws; theories in mathematics, science, medicine, create literature and much more.

Until the invention of the printing press, all documents have been handwritten on different types of physical media, e.g. papyrus, parchment, paper, etc., with copies of documents being made by hand. Printing, however, offered a fast and inexpensive method of reproducing texts that drastically changed the world.

Over time, a vast amount of documents containing information on the history of mankind have been amassed, information that is invaluable for researchers investigating social, political and cultural developments. That being said, this paper covers the information extraction and the targeted Optical Character Recognition (OCR) process, specially designed for the automatic recognition of a Spanish census from the XIX century, registered in printed documents. Once this corpus is transcribed, this paper goes a step forward and introduces a demonstrator model to facilitate researching on this corpus by indexing the data.

Figure 1.1 shows an overview of all the steps involved in the automatic recognition of historical documents, steps which will be covered in detail throughout this paper.

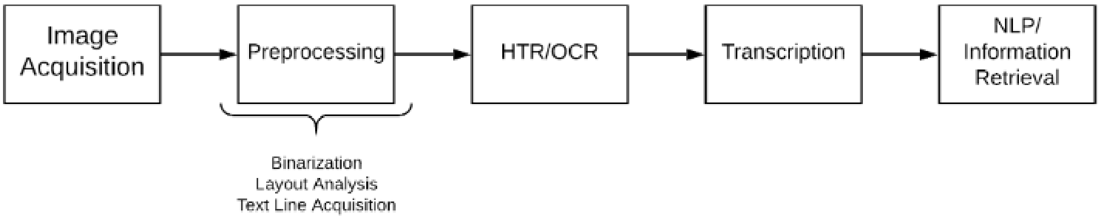


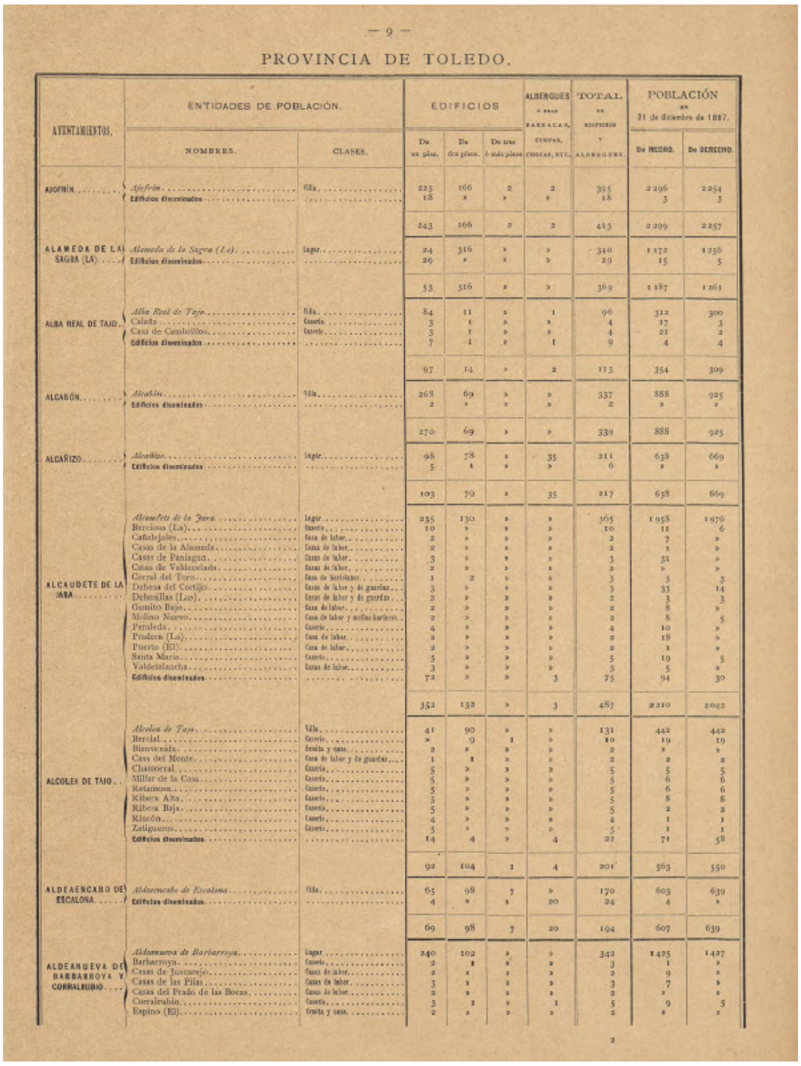
Figure 1.1: The steps in a conventional Historical Document Processing workflow for both handwritten and printed documents, source [1].

1.1 Motivation

Automatic recognition of historical printed texts pose some difficulties that are not present in currently scanned printed text, namely, degraded and smeared paper, paper and typewriters of poor quality, in addition to usual OCR problems. Current OCR problems move around a mixture of layout analysis problems and recognition problems of characters of different styles, fonts, sizes, resolutions, and overlapped characters, just to mention a few. It is important to remark that if the printed characters can be easily isolated, many of these problems are alleviated for recognition processes since the problem becomes a classification problem. If the characters can not be easily isolated and/or the layout is not adequately solved, then some problems arise. This is the case, for example, for thousands of historical newspaper collections or historical forms and records collections residing in archives and libraries, among many other documents.

For the layout analysis problem, robust techniques that are based on deep and convolutional neural networks are being developed [3]. For the recognition of printed characters, the problem can be alleviated through linguistic resources used in the recognition process, namely, language models and vocabularies. However, linguistic resources are not always available for all languages and all tasks. The automatic recognition results can be improved if the task at hand has redundant information about the layout or the content of the documents, information that can help detect and correct errors. This is the situation for the documents that are researched in this project.

This paper introduces a typical collection of historical printed texts that are related to a Spanish census from 1887. This census is researched in the context of the ESPAREL project,[1] that studies the population evolution in the last two centuries and intends to georeference all cities and towns of that period. Fig. 1.2 shows a portion of one page of this collection. The collection is invaluable and could be



— 9 —

PROVINCIA DE TOLEDO.

AYUNTAMIENTOS.	ENTIDADES DE POBLACIÓN.		EDIFICIOS			ALBERGUES ó gran barracas, cuevas, chozas, etc.	TOTAL de edificios y albergues.	POBLACIÓN en 31 de diciembre de 1887.	
	NOMBRES.	CLASES.	De un piso.	De dos pisos.	De tres ó más pisos			De HECHO.	De DERECHO.
AJOFRÍN.	*Ajofrín*	Villa.	225	166	2	2	395	2 296	2 254
	Edificios diseminados		18	»	»	»	18	3	3
			243	166	2	2	413	2 299	2 257
ALAMEDA DE LA SAGRA (LA).	*Alameda de la Sagra (La)*	Lugar.	24	316	»	»	340	1 172	1 256
	Edificios diseminados		29	»	»	»	29	15	5
			53	316	»	»	369	1 187	1 261
ALBA REAL DE TAJO.	*Alba Real de Tajo*	Villa.	84	11	»	1	96	312	300
	Calada	Caserío.	3	1	»	»	4	17	3
	Casa de Cambrillos	Caserío.	3	1	»	»	4	21	2
	Edificios diseminados		7	1	»	1	9	4	4
			97	14	»	2	113	354	309
ALCABÓN.	*Alcabón*	Villa.	268	69	»	»	337	888	925
	Edificios diseminados		2	»	»	»	2	»	»
			270	69	»	»	339	888	925
ALCAÑIZO.	*Alcañizo*	Lugar.	98	78	»	35	211	638	669
	Edificios diseminados		5	1	»	»	6	»	»
			103	79	»	35	217	638	669
ALCAUDETE DE LA JARA.	*Alcaudete de la Jara*	Lugar.	235	130	»	»	365	1 958	1 976
	Berciosa (La)	Caserío.	10	»	»	»	10	11	6
	Cañalejales	Casa de labor.	2	»	»	»	2	7	»
	Casas de la Alameda	Casas de labor.	2	»	»	»	2	1	»
	Casas de Paniagua	Casas de labor.	3	»	»	»	3	31	»
	Casas de Valdeceñada	Casas de labor.	2	»	»	»	2	»	»
	Corral del Toro	Casa de hortelanos.	1	2	»	»	3	3	3
	Dehesa del Cortijo	Casas de labor y de guardas.	3	»	»	»	3	33	14
	Dehesillas (Las)	Casas de labor y de guardas.	2	»	»	»	2	3	3
	Gamito Bajo	Casa de labor.	2	»	»	»	2	8	»
	Molino Nuevo	Casa de labor y molino harinero.	2	»	»	»	2	8	5
	Peraleda	Caserío.	4	»	»	»	4	10	»
	Pradera (La)	Casa de labor.	2	»	»	»	2	18	»
	Puerto (El)	Casa de labor.	2	»	»	»	2	1	»
	Santa María	Caserío.	5	»	»	»	5	19	5
	Valdelalancha	Casas de labor.	3	»	»	»	3	5	»
	Edificios diseminados		72	»	»	3	75	94	30
			352	132	»	3	487	2 210	2 042
ALCOLEA DE TAJO.	*Alcolea de Tajo*	Villa.	41	90	»	»	131	442	442
	Bercial	Caserío.	»	9	1	»	10	19	19
	Bienvenida	Ermita y casa.	2	»	»	»	2	»	»
	Casa del Monte	Casa de labor y de guardas.	1	1	»	»	2	2	2
	Chamorral	Caserío.	5	»	»	»	5	5	5
	Millar de la Casa	Caserío.	5	»	»	»	5	6	6
	Retamosa	Caserío.	5	»	»	»	5	6	6
	Ribera Alta	Caserío.	5	»	»	»	5	8	8
	Ribera Baja	Caserío.	5	»	»	»	5	2	2
	Rincón	Caserío.	4	»	»	»	4	1	1
	Zatigueros	Caserío.	5	»	»	»	5	1	1
	Edificios diseminados		14	4	»	4	22	71	58
			92	104	1	4	201	563	550
ALDEAENCABO DE ESCALONA.	*Aldeaencabo de Escalona*	Villa.	65	98	7	»	170	603	639
	Edificios diseminados		4	»	»	20	24	4	»
			69	98	7	20	194	607	639
ALDEANUEVA DE BARBARROYA Y CORRALRUBIO.	*Aldeanueva de Barbarroya*	Lugar.	240	102	»	»	342	1 425	1 427
	Barbarroya	Caserío.	2	1	»	»	3	1	»
	Casas de Juncarejo	Casas de labor.	2	»	»	»	2	9	»
	Casas de las Pilas	Casas de labor.	3	»	»	»	3	7	»
	Casas del Prado de las Bocas	Casas de labor.	2	»	»	»	2	»	»
	Corralrubio	Caserío.	3	1	»	1	5	9	5
	Espino (El)	Ermita y casa.	2	»	»	»	2	»	»

2

Figure 1.2: Example image of a page corresponding to the first page of the Toledo province.

of interest to various parties as it contains information about demographic factors, economic evolution, geographical and reference information, and migration movements.[2]

[1] https://www.fbbva.es/equipo/esparel-una-plataforma-digital-para-explorar-la-transicion-del-ant

[2] https://www.ine.es/inebaseweb/treeNavigation.do?tn=193862

This type of documents is quite common in all countries and for different situations: border registers, bank accounts, financial records, numerical data related with a record of daily weather[3], etc. In many collections of this type, the main goal is to move the tabular information to a database to research a specific problem. Given these documents from the ESPAREL project, like the one that Fig. 1.2 shows, the very attractive goal is to retrieve the information without errors. However, this goal is hard to attain due to the fact that physical documents suffer from the usual problems that afflict historical documents, that is: paper degradation, smear, noise, layout problem, etc. This paper describes the data extraction process for this collection and the techniques that were used to overcome the aforementioned problems and the methodology applied to get the information practically without errors by making use of the redundant information present in these documents.

The automatic recognition of these images using conventional OCR systems do not provide very good results and the extracted information ends up being useless unless it is manually reviewed. Therefore specific OCR training can provide much better results. This opens a new problem which is to get ground truth data for training the OCR system. A remarkable problem with these types of documents (census, numerical tables, etc) is that linguistic resources are scarce and/or out-of-date. Therefore, the restrictions that can provide a language model or a vocabulary to improve recognition results can not be used. Consequently, to mitigate the absence of linguistic resources, other types of restrictions have to be used, like the redundant information that these tables could contain. This is the solution that was adopted in this research.

Given the importance and interest of this Spanish census for conducting demographic studies, this project also covers the indexing of the data concluding with the construction of a demonstrator search engine model to facilitate the research on this corpus of documents. In order to tackle the indexing and searching of the data, a technique that uses Probabilistic Indexes (PI) will be used as introduced in [4]. This technique has been demonstrated [5] to be highly efficient

[3] https://brohan.org/Google-Vision/OCR-weatherrescue/months.html

for searching information in massive collections of documents, with numerous demonstrator search engines[4] that already employ it.

1.2 Objectives

The main objective of this project is the transcription of the Spanish census of 1887 from the ESPAREL collection, relying only on redundant information contained in this collection without using any linguistic resources.

The partial objectives derived from this main objective, are as follows:

1. Separate the data into three categories: i) to be extracted; ii) redundant; iii) and unnecessary data.
2. Preprocess the data, applying computer vision techniques for correcting the document images as needed, in order to correctly extract the lines from the tabular data to be transcribed.
3. Implement a targeted OCR system based on the current technology of combining Convolutional Neural Networks with Bi-directional Long-Short Term Memory networks, to be trained with and transcribe the data resulting from the previous preprocessing phase.
4. Detect and correct the transcription errors.

As a secondary objective of this project, is the construction of a demonstrator search engine model for retrieving information from the transcribed census by means of queries. This secondary objective can be divided into partial objectives as follows:

5. Extract probabilistic information for indexing the ESPAREL corpus.
6. Design and implement a probabilistic indexing-based search engine for the retrieval of information from the corpus.

[4] http://carabela.prhlt.upv.es/en/demonstrators

1.3 Paper Structure

In order to aid the evaluation of the objectives defined in this project and the degree with which they have been achieved, this paper has been divided into several chapters. This division can be used as a guide of the necessary steps for replicating the experiments presented in this paper and also as an approach for the automatic recognition of historical documents.

After an initial analysis of the motivation behind this project in chapter 1 and the related work in chapter 2, this paper presents the ESPAREL corpus with the characteristics and structure of the data set (chapter 3), detailing both the information to be extracted, as well as the information that gives support to the detection and correction of errors.

In chapter 4 we detail the preprocessing of the ESPAREL data, with the necessary steps in order to guarantee the correction of the line extraction process. Once this process is explained, chapter 5 continues with the description of the resulting data and the construction of a model for a targeted OCR system. In this chapter, the neural model employed is presented, together with the steps needed for training the system to transcribe the data.

In chapter 6, the results of the transcription process are analyzed and the process of error detection and corrections is explained. This chapter concludes with the information extraction of the relevant data. Chapter 7 introduces the theoretical framework for indexing the data that gives support to building a demonstrator search engine to conduct queries on the extracted information, also explained in this chapter.

Finally, this paper concludes with chapter 8 with a summary of the key points of this project. In addition, we detail the software requirements for replicating the experiments presented in this paper (Appendix A).

CHAPTER 2

Related Work

Automatic recognition of historical documents is a very active field [6], [7], [8] with many collections having been transcribed and made available to the public. However, an enormous amount of documents, scanned and stored as image data, still remain to be transcribed in libraries and archives all over the world. While there are many OCR products that tackle the automatic recognition of modern documents, when dealing with printed historical documents there are characteristics of the documents, such as ink splodges, degradation of the pages and out-of-use characters and fonts, etc..., that make the automatic recognition a very challenging task for off the shelf OCR products. In order to address these problems, a layout analysis of the data under optimal conditions needs to be done. As a result of this process, the lines to be transcribed are correctly extracted and prepared for training a targeted OCR system, specially designed for the task at hand. With this in mind, this section explores the state of the art of the:

1. document layout analysis process;
2. OCR systems for transcribing document images;
3. indexing of documents and Information Retrieval systems.

2.1 Document Layout Analysis

The document layout analysis process has the purpose of identifying regions of interest in the document images. According to [9], there are three types of layout analysis methods: 1) block based; 2) pixel based; 3) and connected components based classification of regions of the documents as text, images or tables. Given that the ESPAREL data to be transcribed is organized in tables, as briefly mentioned in chapter 1 and more thoroughly explained in chapter 3, we focused only on block based layout analysis techniques. For this task, Convolutional Neural Networks (CNN) are widely adopted [10], [11], [12] considering their expressivity and capacity for learning image features. However, neural network approaches require labeled data to train the models. Other approaches make use of the geometric characteristics of the documents to segment the images into regions of interests (lines) based on projection profile cuts [13], [14]. Considering that the ESPAREL data is organized into tables with a highly regular layout, this latter approach of geometric layout analysis was used to segment the tables into lines.

2.2 OCR Systems

With the layout analysis techniques explored, in the next step we'll investigate the state of the art for OCR models. The task of text recognition is split into two categories: constrained or unconstrained recognition. The task of constrained text recognition relies on having a fixed lexicon or dictionary which restricts the possible words that can be recognized, greatly helping to improve the error rate of the system. However, for corpora like the one presented in this paper, composed of tables that contain mainly numbers and proper names, there are no language models available. With this in mind, we focused only on unconstrained text recognition systems.

A widely used approach to the unconstrained text recognition task is the use of Hidden Markov Models (HMMs). HMMs have been successfully employed for modeling time-varying signals in both speech-recognition and handwriting recognition tasks. HMM based systems can either employ an implicit segmentation of the word images by representing them as a sequence of observations [15], or an explicit segmentation to split words into letters or pseudoletters [16], [17]. However, the HMM based systems assume the probability of each observation depends only on the current state, with contextual dependencies difficult to model as pointed out in [18]. Therefore, the authors of [18] propose a recurrent neural network (RNN)-based approach for sequence modelling. In this system, the recognition process takes into account both the previous context and the future context through the use of Bidirectional Long Short Term Memory networks (BLSTM). To avoid the pre-segmentation of the data, the system employs a Connectionist Temporal Classification (CTC) output layer, which produces a probability distribution over the character transcriptions without having to align the input time steps with their corresponding output transcription. Following this approach, more recent papers explore the combination of deep convolutional networks for image encoding and LSTMs for language modeling [19], [20], [21]. Given the very good results reported in these papers, with improvements over the models that only employ LSTMs, the same approach of CNN + LSTMs based OCR system was used for this project.

2.3 Indexing and Information Retrieval Systems

As stated before, a secondary objective set in this project is the creation of a demonstrator search engine for retrieving information from the ESPAREL corpus. In order to build an Information Retrieval system to search textual information in images of the documents, these images need to be indexed based on their content.

In the field of handwritten text recognition (HTR), the Indexing and Searching problem can be addressed using Probabilistic Indexes (PI) [4], a technology that has been developed by the PRHLT research center. This approach is based on techniques known as Key Word Spotting (KWS), which can be used to annotate each image with information on words that have a probability to appear in it, along with their corresponding probability and positions. This information is generated for each image in the form of a heat map where each pixel indicates the probability that it belongs to one or more word hypotheses.

The main advantage of using PI lies in the fact that they capture the uncertainty inherent to handwritten text recognition, saving the n-best possible interpretations of the words to be recognized. In this way, the search for information in text images is much more robust than using only the automatic transcription of a document. This aspect, together with the fact that PI-based indexing solutions allow a very efficient search process in massive collections of documents [5], providing control over the relevance of the results through the confidence levels of the word-graphs interpretations [21], is what motivated the use of PI for building a demonstrator search engine.

In [21], besides using PI for multiple keywords queries, employing a combination of AND, OR, NOT of the results supplied by the individual terms of the query, the paper provides a solution for structured multi-word queries, for information retrieval of tabular data from text images. By allowing users to form structured queries for retrieving information based on the columns of the tabular data, the relevance of the results is greatly increased, facilitating the research on this corpus of documents. This is a very attractive aspect which will be explored for the search engine to be built in this project.

CHAPTER 3

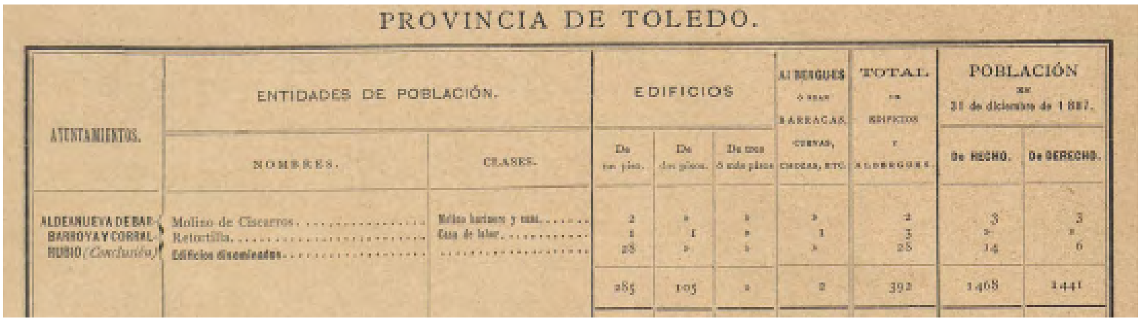
Task Description

The ESPAREL collection is a Spanish census from 1887 that is printed in tabular format and owned by Biblioteca Nacional de España.[1] The collection is composed of eight volumes and it comprises about 1 500 double page scanned images, that is, more than 3 000 page images. The collection is organized in provinces (49 in Spain in that period) sorted in alphabetical order. The information in each province is, in turn, organized in cities (or municipalities) and towns sorted in alphabetical order as Fig. 1.2 shows.

The meaning of each column is written in the header of each page, and it is consistent along all pages. Fig. 3.1 shows a detailed header. Column "AYUNTAMIENTO" (town-hall) refers to the whole city or town name. Second column is the place name (suburbs or other place names). Column "CLASES" refers to the type of place: farmer house, small village, mill, etc. Column "EDIFICIOS" is the number of living buildings with one, two, or three or more floors. Column "ALBERGUES" is the number of hostels. Column "POBLACION" is the amount of people that were living in that place ("DE HECHO") or registered in that place ("DE DERECHO").

Note that for each place, the total number of buildings is registered in the "TOTAL" column. The last rows of each city or town includes the totals for the previous rows. This row is remarked with a bold multicolumn line. A double quote is used to denote a null value. These accumulated values can be used as an error-correcting code to locate misrecognition in previous rows and columns.

[1] http://bdh.bne.es/bnesearch/detalle/bdh0000199638

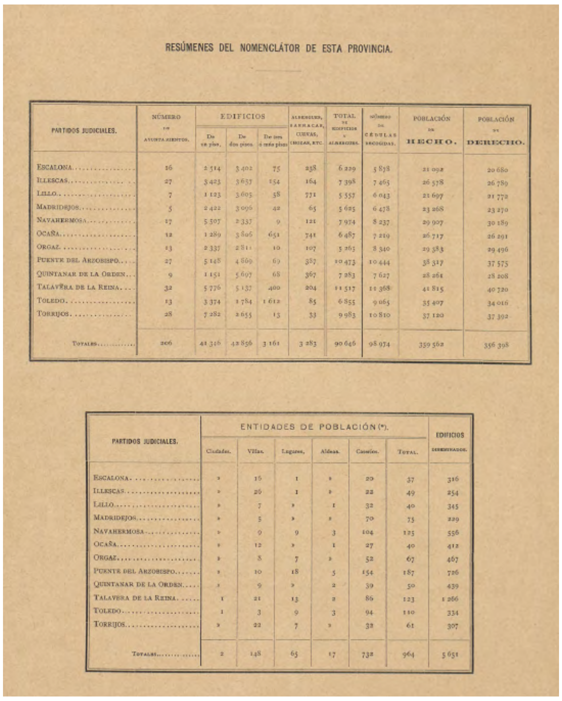


PROVINCIA DE TOLEDO.

AYUNTAMIENTOS.	ENTIDADES DE POBLACIÓN.		EDIFICIOS			ALBERGUES ó sean BARRACAS, CUEVAS, CHOZAS, ETC.	TOTAL DE EDIFICIOS Y ALBERGUES.	POBLACIÓN EN 31 de diciembre de 1887.	
	NOMBRES.	CLASES.	De un piso.	De dos pisos.	De tres ó más pisos.			De HECHO.	De DERECHO.
ALDEANUEVA DE BARBARROYA Y CORRAL-RUBIO (*Conclusión*)	Molino de Ciscarros	Molino harinero y casa	2	»	»	»	2	3	3
	Retortilla	Casa de labor	1	1	»	1	3	»	»
	Edificios diseminados		28	»	»	»	28	14	6
			285	105	»	2	392	1468	1441

Figure 3.1: Detailed view of the header of a page.

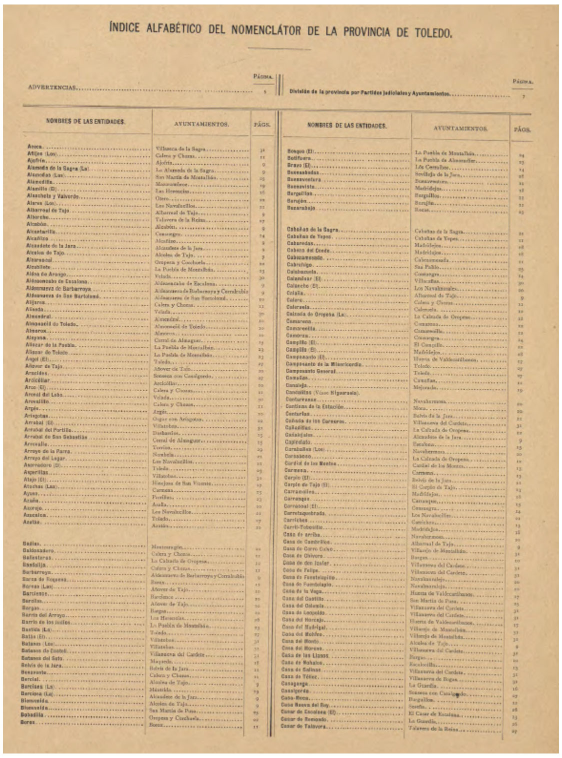
At the end of every province, there are several summaries: the total for the province (see the left image of Fig. 3.2) and a sorted list of all places in each province that indicate which town or city belongs to. The former information was not used in the research because it was not necessary. Fig. 3.2 shows an example of the latter information. This information was very useful to disambiguate places with the same name that belonged to different cities or towns. Note that this is the kind of restriction that can be very useful to restrict the search space and to improve recognition results as we mentioned in the Introduction chapter.



RESÚMENES DEL NOMENCLÁTOR DE ESTA PROVINCIA.

PARTIDOS JUDICIALES.	NÚMERO DE AYUNTAMIENTOS.	EDIFICIOS			ALBERGUES, BARRACAS, CUEVAS, CHOZAS, ETC.	TOTAL DE EDIFICIOS Y ALBERGUES.	NÚMERO DE CÉDULAS RECOGIDAS.	POBLACIÓN DE HECHO.	POBLACIÓN DE DERECHO.
		De un piso.	De dos pisos.	De tres ó más pisos.					
ESCALONA	16	2 514	3 402	75	238	6 229	5 878	21 092	20 680
ILLESCAS	27	3 423	3 657	154	164	7 398	7 465	26 578	26 789
LILLO	7	1 123	3 605	58	771	5 557	6 043	21 697	21 772
MADRIDEJOS	5	2 422	3 096	42	65	5 625	6 478	23 268	23 270
NAVAHERMOSA	17	5 507	2 337	9	121	7 974	8 237	29 907	30 189
OCAÑA	12	1 289	3 806	651	741	6 487	7 219	26 717	26 291
ORGAZ	13	2 337	2 811	10	107	5 265	8 340	29 383	29 496
PUENTE DEL ARZOBISPO	27	5 148	4 869	69	387	10 473	10 444	38 317	37 575
QUINTANAR DE LA ORDEN	9	1 151	5 697	68	367	7 283	7 627	28 261	28 208
TALAVERA DE LA REINA	32	5 776	5 137	400	204	11 517	11 368	41 815	40 720
TOLEDO	13	3 374	1 784	1 612	85	6 855	9 065	35 407	34 016
TORRIJOS	28	7 282	2 655	13	33	9 983	10 810	37 120	37 392
TOTALES	206	41 346	42 856	3 161	3 283	90 646	98 974	359 562	356 398

PARTIDOS JUDICIALES.	ENTIDADES DE POBLACIÓN (*).						EDIFICIOS DISEMINADOS.
	Ciudades.	Villas.	Lugares.	Aldeas.	Caseríos.	TOTAL.	
ESCALONA	»	16	1	»	20	37	316
ILLESCAS	»	26	1	»	22	49	254
LILLO	»	7	»	1	32	40	345
MADRIDEJOS	»	5	»	»	70	75	229
NAVAHERMOSA	»	9	9	3	104	125	556
OCAÑA	»	12	»	1	27	40	412
ORGAZ	»	8	7	»	52	67	467
PUENTE DEL ARZOBISPO	»	10	18	5	154	187	726
QUINTANAR DE LA ORDEN	»	9	»	2	39	50	439
TALAVERA DE LA REINA	1	21	13	2	86	123	1 266
TOLEDO	1	3	9	3	94	110	334
TORRIJOS	»	22	7	»	32	61	307
TOTALES	2	148	65	17	732	964	5 651



ÍNDICE ALFABÉTICO DEL NOMENCLÁTOR DE LA PROVINCIA DE TOLEDO.

ADVERTENCIAS........ Página 5 || División de la provincia por Partidos judiciales y Ayuntamientos........ Página 7

NOMBRES DE LAS ENTIDADES.	AYUNTAMIENTOS.	PÁGS.	NOMBRES DE LAS ENTIDADES.	AYUNTAMIENTOS.	PÁGS.

Figure 3.2: Left side image shows an example of the summary page for a province. Right side image shows an example of the index of places for a province.

To facilitate the posterior preprocessing phase, we divided the data into three categories: i) to be extracted; ii) redundant; iii) and unnecessary data.

The first category consists of the tabular data shown in Figure 1.2. The automatic recognition of this data, as detailed in the beginning of this chapter, is the main focus of this project.

The second category consists of sorted lists of all places in each province. Even though this information is also captured in the nested structure of the tabular data of the first category, as mentioned before this redundant data is very useful disambiguate between places with the same name but different locations.

The last category consists of data such as summaries for each province, pages of notices about the layout of the documents or certain cultural details such as the manner in which some building were constructed, and the administrative division of territories which are also compiled in the indices of places for each province, as shown in the right side of Figure 3.2. This information was considered unnecessary and was removed from the set of images to be transcribed.

While the first two categories of data both need to be transcribed, with the third category removed, in the end all the information needed for conducting researches on the ESPAREL corpus is gathered in the first category. This data was later indexed in order to construct a query based search mechanism for the retrieval of information.

CHAPTER 4

Data Preprocessing

As stated before, the document layout analysis process was a necessary step for identifying the regions of interest in the document images, prepare and extract the data for the recognition process. Given that the images consist of tabular data with the information of the rows independent from one another, the data was split into lines as explained hereafter.

Even though the data was organized into tables that allowed for easy processing, the line extraction process had to overcome several obstacles such as page skew, extracting data while keeping the hierarchical divisions, stains on the pages making line extraction more difficult, municipality data breaking across several pages and so on. This section will explore all these obstacles along with the solutions identified. In order to address these obstacles, this process has been divided into the following steps:

1. Preprocessing the images to correct the skew.
2. Removing irrelevant data by detecting and isolating the layout of the tables.
3. Splitting the tables into smaller sections that could be processed independently.
4. Detecting and extracting rows (lines) for the transcription process.
5. Dealing with nested groups of information.

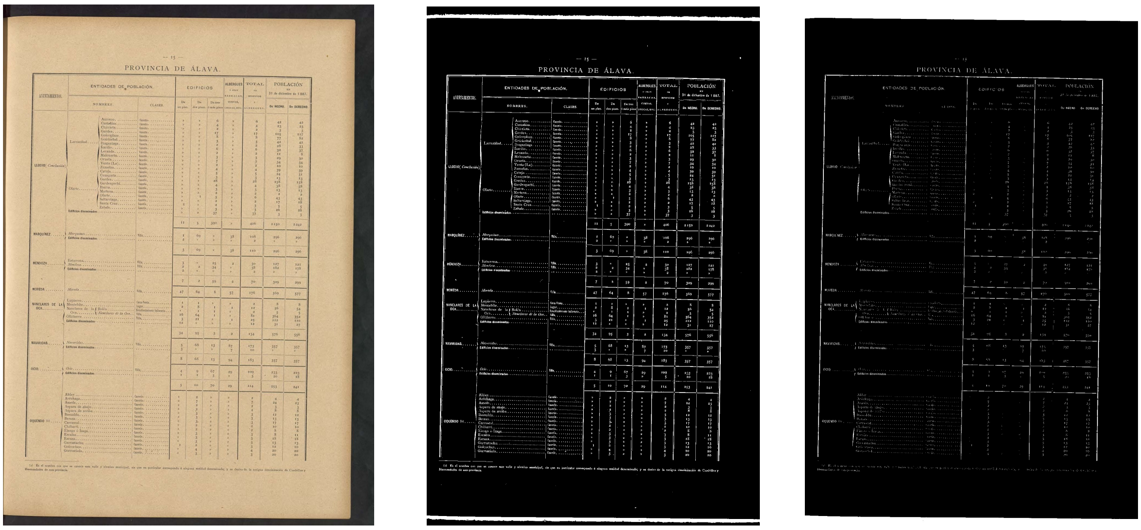
4.1 Skew correction

Considering that all eight volumes we processed were available in scanned format with each page corresponding to two consecutive pages, the first step was to split the scanned images into two halves corresponding to the left and right pages. Because the scanned images covered two pages, each half had a certain degree of warping and skew towards the book-binding. Therefore, the skew correction process was a crucial step to enhance the quality of the results of the OCR system.

Once each image was divided into halves we could address the skew by following an approach similar to the one presented in [22], that is, compute the Progressive Probabilistic Hough Transform (PPHT) to detect vertical and horizontal image lines. Alternative methods were also used, where the document images were rotated under multiple degrees, with projection profile techniques applied for identifying the correct skew angle. This was done by maximizing the number of vertical/horizontal pixels corresponding to the outer vertical/horizontal lines of the tables, which indicated the optimum rotation degree. However, projection profile based methods are very computational intensive given that the images have to be analyzed under multiple degrees of rotation. For this reason, these methods were discarded in favor of the PPHT technique.

Before applying PPHT to detect the vertical and horizontal lines of the tables, the document images had to be binarized. In OpenCV, which is the library that was used throughout the data preprocessing phase, there are a number of algorithms implemented for binarizing an image. In standard binarization techniques, a global threshold is applied over the values of the pixels, with pixels that fall below the threshold being converted to a value of 0, and those above to a maximum value set by the user. Instead of manually selecting the global threshold, we used Otsu's binarization method which determines the optimal global threshold value from the image histogram [23].

However, this technique is not sufficiently robust for historical document images given that the pages suffer from variable degrees of discoloration caused by aging of the paper, which gives rise to loss of information. Therefore, an adaptive thresholding technique is more appropriate when working with images with varying illumination. This technique applies different thresholds to different regions based on the values of the pixels of that region. Figure 4.1 illustrates the difference between the results of applying global or adaptive thresholding techniques for document images with varying degree of illumination. As it can be seen, for the image documents of the ESPAREL corpus the Adaptive Thresholding technique retains more information than Global Thresholding technique. This makes the process of detection of lines more robust.



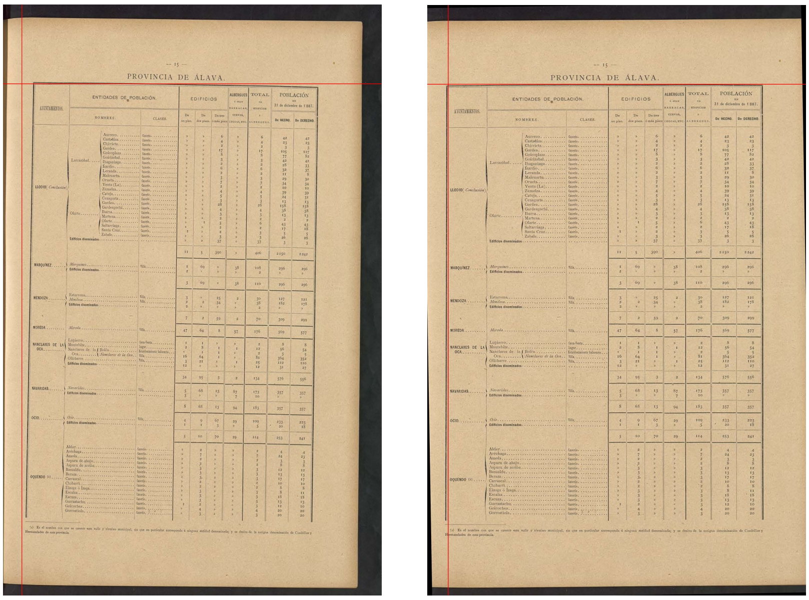
(a) Original page. **(b)** Adaptive Thresholding. **(c)** Global Thresholding.

Figure 4.1: Difference between global and local binarization techniques.

With the images binarized, the next step was to apply PPHT to detect vertical and horizontal image lines. The PPHT method has a series of parameters that need to be set: degree precision, minimum pixel length of the lines to be detected and the gap between pixels forming a line, among others. In order to compute the skew angle of the images, the skeleton lines forming the table layout were considered to be more appropriate than detecting the lines formed by the words present in the tables. For this reason, the gap between the line pixels was set with a value that was half the average space between the rows of the table and with a minimum line length sufficiently high in order to ignore possible lines created by joining characters together. With the lines detected, the next step was to calculate the slope of the vertical and horizontal line of maximum length. With this approach we detected three possible cases of deviation:

1. Similar horizontal and vertical slope.
2. Horizontal slope different from 0 and vertical slope close to 0.
3. Horizontal slope close to 0 and vertical slope different from 0.

While the first case indicated a classic skew of the image, possibly due to incorrect scanning alignment, the other two cases indicated a degree of warping of the image besides a certain skew. Once this measurement had been done, we determined that the warping present only affected a small area on the interior of the page. However, when correcting the skew by applying an inverse rotation based on the average of the angles of the two slopes, the effects of the warping were partly mitigated. Figure 4.2 shows the results of this skew correction method. It is important to remark that the process of rotating an image incurs in a loss of information which was mitigated by using a bicubic interpolation over a 4x4 pixel neighborhood [1].



(a) Original page.

(b) Skew corrected page.

Figure 4.2: Results of the skew correction process for the page shown in Figure 4.1a, with red guidelines to better illustrate the skew present.

[1] https://medium.com/hd-pro/bicubic-interpolation-techniques-for-digital-imaging-7c6d86dc35dc

4.2 Layout Detection and Data Organization

With the skew corrected, the next step of the process was to isolate the data that needed to be transcribed. This implied both the detection of the outer layout of the tables and the detection of the inner layout of the tables in order to capture the hierarchical structure of the data. In all there are a maximum of four levels of hierarchical data of which one was already addressed, with another being dealt with in this section, and the last two addressed in section 4.4. These four levels are shown in Figure 4.3.

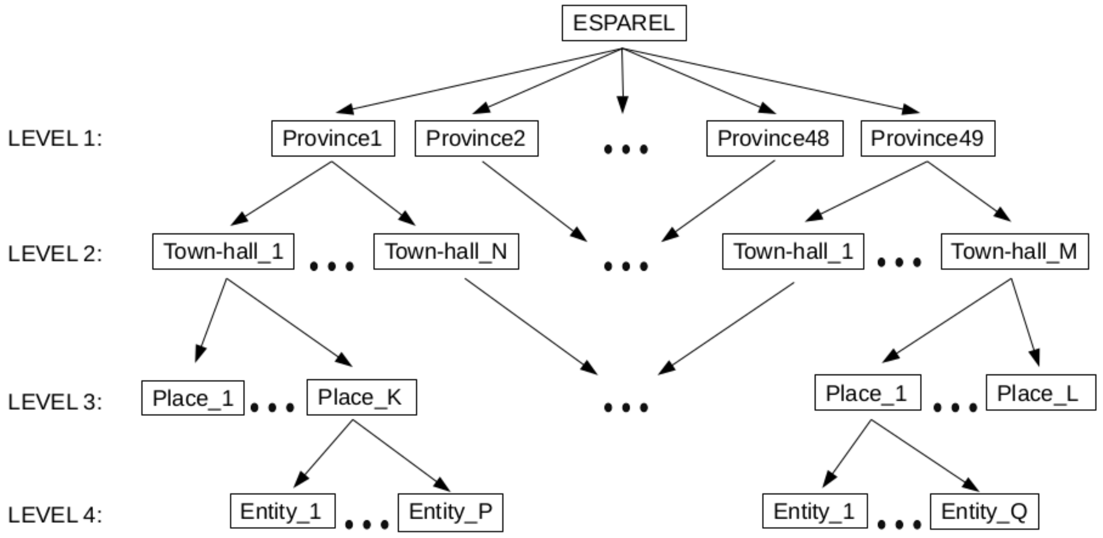
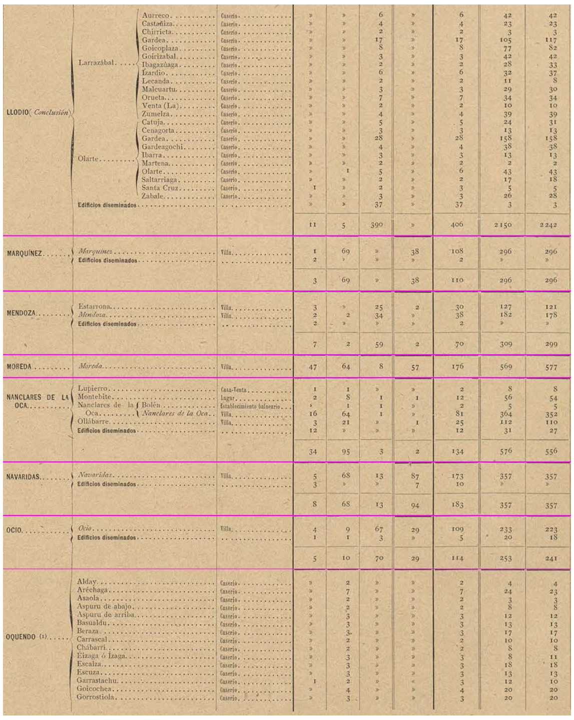


Figure 4.3: The division of the ESPAREL data into four administrative levels, with the fourth level not always present.

The first hierarchical level was covered in chapter 3, when the eight volumes of the ESPAREL corpus had been organized into 49 provinces, where the images of each province had been manually separated.

In order to address the next hierarchical level, that is divide the province tables into town-halls, the outer layout of the table had to be detected. This task was done by applying morphology operations of dilation followed by erosion in order to eliminate all the characters from the pages, leaving only the skeletal layout of the table. In a following step, the outer layout of the table was detected by applying projection profile methods both vertically and horizontally. This was then used to crop the images, focusing only on the relevant information, thus removing unnecessary data such as titles, footnotes, or page numbers that indicated how the pages were organized into provinces. The headers of the tables were also removed as they were repeated at the top of each page and contained the same structure, dimension and information, as shown in Fig. 3.1.

With the tables cropped, the next step was to divide the data according to town-halls. This was done by making use of the fact that every town-hall ended with a "TOTAL" row, separated from the previous rows by a horizontal line, and from the next town-hall data by a thicker horizontal line. By using a combination of Gaussian filters and projection profile techniques, these thicker horizontal lines could be isolated from the rest of the data, and used as cutting points to divide the tables into town-halls. The results of this process can be seen in Figure 4.4.



LLODIO (*Conclusión*)	Larrazábal	Aurreco	Caserío	»	»	6	»	6	42	42
		Castañiza	Caserío	»	»	4	»	4	23	23
		Chirrieta	Caserío	»	»	2	»	2	3	3
		Gardea	Caserío	»	»	17	»	17	105	117
		Goicoplaza	Caserío	»	»	8	»	8	77	82
		Goirizabal	Caserío	»	»	3	»	3	42	42
		Ibagazúaga	Caserío	»	»	2	»	2	28	33
		Izardio	Caserío	»	»	6	»	6	32	37
		Lecanda	Caserío	»	»	2	»	2	11	8
		Malcuartu	Caserío	»	»	3	»	3	29	30
		Orueta	Caserío	»	»	7	»	7	34	34
		Venta (La)	Caserío	»	»	2	»	2	10	10
		Zumelza	Caserío	»	»	4	»	4	39	39
	Olarte	Catuja	Caserío	»	»	5	»	5	24	31
		Cenagorta	Caserío	»	»	3	»	3	13	13
		Gardea	Caserío	»	»	28	»	28	158	158
		Gardeagochi	Caserío	»	»	4	»	4	38	38
		Ibarra	Caserío	»	»	3	»	3	13	13
		Martena	Caserío	»	»	2	»	2	2	2
		Olarte	Caserío	»	1	5	»	6	43	43
		Saltarriaga	Caserío	»	»	2	»	2	17	18
		Santa Cruz	Caserío	1	»	2	»	3	5	5
		Zabale	Caserío	»	»	3	»	3	26	28
	Edificios diseminados			»	»	37	»	37	3	3
				11	5	390	»	406	2150	2242
MARQUÍNEZ	*Marquínez*		Villa	1	69	»	38	108	296	296
	Edificios diseminados			2	»	»	»	2	»	»
				3	69	»	38	110	296	296
MENDOZA	Estarrona		Villa	3	»	25	2	30	127	121
	Mendoza		Villa	2	2	34	»	38	182	178
	Edificios diseminados			2	»	»	»	2	»	»
				7	2	59	2	70	309	299
MOREDA	*Moreda*		Villa	47	64	8	57	176	569	577
NANCLARES DE LA OCA	Lupierro		Casa-Venta	1	1	»	»	2	8	8
	Montebite		Lugar	2	8	1	1	12	56	54
	Nanclares de la Oca	Bolén	Establecimiento balneario	»	1	1	»	2	5	5
		Nanclares de la Oca	Villa	16	64	1	»	81	364	352
	Olllábarre		Villa	3	21	»	1	25	112	110
	Edificios diseminados			12	»	»	»	12	31	27
				34	95	3	2	134	576	556
NAVARIDAS	*Navaridas*		Villa	5	68	13	87	173	357	357
	Edificios diseminados			3	»	»	7	10	»	»
				8	68	13	94	183	357	357
OCIO	*Ocio*		Villa	4	9	67	29	109	233	223
	Edificios diseminados			1	1	3	»	5	20	18
				5	10	70	29	114	253	241
OQUENDO (1)	Alday		Caserío	»	2	»	»	2	4	4
	Aréchaga		Caserío	»	7	»	»	7	24	23
	Asaola		Caserío	»	2	»	»	2	3	3
	Aspuru de abajo		Caserío	»	2	»	»	2	8	8
	Aspuru de arriba		Caserío	»	3	»	»	3	12	12
	Basualdu		Caserío	»	3	»	»	3	13	13
	Beraza		Caserío	»	3	»	»	3	17	17
	Carrascal		Caserío	»	2	»	»	2	10	10
	Chábarri		Caserío	»	2	»	»	2	8	8
	Éizaga ó Ízaga		Caserío	»	3	»	»	3	8	11
	Escalza		Caserío	»	3	»	»	3	18	18
	Escuza		Caserío	»	3	»	»	3	13	13
	Garrastachu		Caserío	1	2	»	«	3	12	10
	Goicochea		Caserío	»	4	»	»	4	20	20
	Gorrostiola		Caserío	»	3	»	»	3	20	20

Figure 4.4: Cropped table from the page shown in Figure 4.2b, in which the town-halls were split by the ending "TOTAL" line.

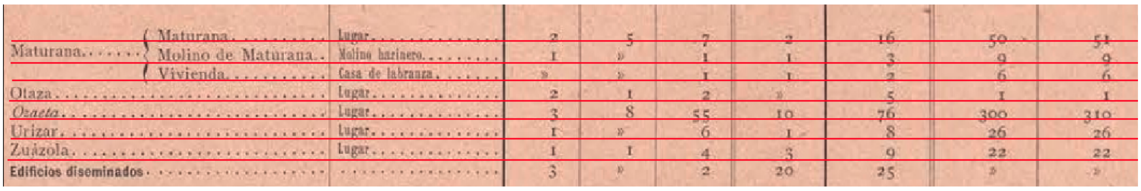
It is important to mention that as it can be seen in Figure 4.4, there are town-halls that extend over more than one page. This case was easily solved by processing the images in a sequential manner, where sub-tables that did not end with a "TOTAL" line were concatenated to subtables from the following pages until a "TOTAL" line was detected.

Following this approach, the ESPAREL corpus was organized by provinces, then by town-halls. This greatly helped the reconstruction of the hierarchical order of the data, which was crucial for building a database with the transcribed information.

4.3 Line Extraction

With the tables structured by provinces, and now also by town-halls, the next step was to detect and extract individual rows (lines) to be used as input for the transcription system. In order to facilitate the line extraction process, the first column corresponding to the name of the town-hall was separated from the rest of the columns, as its characters were usually not aligned with the rest of the rows. The town-hall names were then cropped, with names split over more than one row divided into lines with projection profile techniques that will be detailed below.

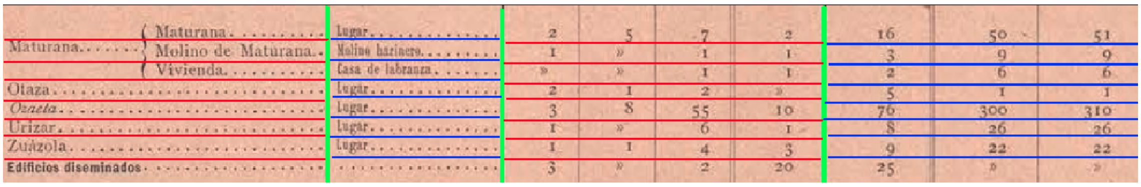
To divide the rest of the columns into rows, the space between the lines was detected by projecting the pixels of the images horizontally and setting a pixel threshold to delimit the rows. As it can be seen in Fig. 4.5, splitting the rows in a single step, covering the entire space from left to right could give rise to many errors as the characters were not perfectly aligned. More so, the vertical lines separating the columns would create interference by raising the threshold value for detecting empty space and thus possibly clipping some characters.



Maturana........	Maturana..........	Lugar...............	2	5	7	2	16	50	51
	Molino de Maturana..	Molino harinero.........	1	»	1	1	3	9	9
	Vivienda..........	Casa de labranza.......	»	»	1	1	2	6	6
Otaza............................		Lugar..............	2	1	2	»	5	1	1
Ozaeta............................		Lugar..............	3	8	55	10	76	300	310
Urizar............................		Lugar..............	1	»	6	1	8	26	26
Zuázola............................		Lugar..............	1	1	4	3	9	22	22
Edificios diseminados..................		....	3	»	2	20	25	»	»

Figure 4.5: Sub-table corresponding to the *Barrundia* city-hall, showing an incorrect separation of rows, clipping characters, caused by character misalignment.

In [14], a proposed solution to this problem would be to divide the image into columns and process each column individually. By working with narrower rows, this technique would avoid clipping characters due to the irregularities of the typesetting process. Following this approach, one solution to this problem was to use the natural separation of the tables into the already defined columns after which the detection of the space between lines was carried out, finalizing with the concatenation of the resulting fragmented rows into a single line. However, we noticed that the columns that were too narrow, due to the reduced pixel count could not be split correctly if imperfections of the images such as stains were present. Fig. 4.6 shows an example of the proposed fragmentation of the table into columns and rows, which decreased the chance of clipping parts of the characters.



Maturana.	Maturana.	Lugar.	2	5	7	2	16	50	51
	Molino de Maturana.	Molino harinero.	1	»	1	1	3	9	9
	Vivienda.	Casa de labranza.	»	»	1	1	2	6	6
Otaza.		Lugar.	2	1	2	»	5	1	1
Ozaeta.		Lugar.	3	8	55	10	76	300	310
Urizar.		Lugar.	1	»	6	1	8	26	26
Zuñzola.		Lugar.	1	1	4	3	9	22	22
Edificios diseminados.			3	»	2	20	25	»	»

Figure 4.6: Sub-table corresponding to the *Barrundia* city-hall, showing a correct separation of rows, by previously dividing the table into four columns to be processed separately.

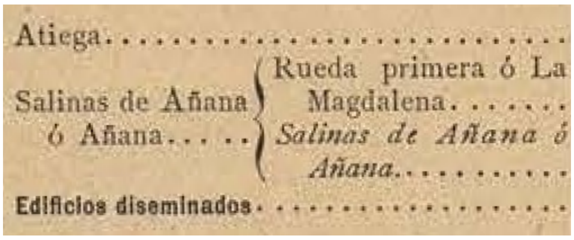
This solution proposed and shown in Fig. 4.6, also gave us the opportunity of detecting incorrect divisions of the tables into rows, given that any mismatch in the number of fragmented rows of the four columns would indicate an error. With this approach, the columns that were not completely split into rows due to imperfections of the images, were passed trough a secondary projection profile filter with a different threshold value for detecting white spaces. While this worked for almost all tables, there were still a handful of conflicting tables that were corrected manually.

4.4 Dealing with Nested Structures

An equally important task besides splitting the tables into rows was making sure that the hierarchy of the population entities would not be lost and thus be able to recreate the administrative divisions needed to build the database. With the first two hierarchical levels already addressed in the previous sections, in this section we'll cover the division of places into entities forming nested structures as seen in Figure 4.7.

One problem caused by the nested structures present in the tables was the case of population named entities that were split over two lines. Separating the table into rows could pose problems for cases such as this because the link between both parts of the named entities would be difficult to reconstruct once the transcription process would be over. This would cause inaccuracies in the database, by having entries with parts of named entities not linked to any numerical data.

The solution to this problem was obtained by having previously separated the tables into columns. This allowed us to detect cases such as these that would otherwise have been obviated with other approaches. It's important to remark that the main reason the named entities were split over two lines, was because of the second level of nesting that reduced the designated space, as shown in Fig. 4.7.



Atiega..........................

Salinas de Añana
ó Añana.....

Rueda primera ó La
Magdalena.......
*Salinas de Añana ó
Añana..........*

Edificios diseminados..................

Figure 4.7: Population entities column showing name entities split over two lines in a nested structure.

An aspect that has not been mentioned before and is of great relevance for piecing together both parts of the split named entities, is that the population entities column had been processed with a low-value threshold that allowed detecting curly brackets that signaled nested structures. This was done to remove the left side of the brackets, leaving only the right side for splitting into rows. As it can be seen in Fig. 4.7, a telltale mark of named entities split over two lines was that the second line always had an indent that could be detected through projection techniques once the left part of the brackets had been removed. Following this approach, the two lines corresponding to a named entity had been pieced together and concatenated with the rows from the rest of the columns.

While removing the left part of curly brackets facilitated the division of the table into rows and allowed detecting split named entities, it eliminated information about the hierarchical order that would be necessary for creating the database. However, this information was duplicated in lists at the end of every province, as shown in the right picture of Fig. 3.2. These lists were sorted in alphabetical order, exactly in the same manner as the population entities column, which after a similar line extraction and transcription process, allowed recovery of the deleted information.

CHAPTER 5

Data Recognition

With the data preprocessed and organized by hierarchical levels, this chapter will present an overview of the extracted data to be transcribed, in addition to describing the process of the construction of an OCR system specially designed for transcribing the ESPAREL dataset.

5.1 ESPAREL dataset

The information required in the ESPAREL project was contained in lines such as the ones that are shown in Fig. 1.2. The characteristics of this dataset, with the distribution of the lines between the various cities, can be seen in Table 5.1. Table 5.1 shows that there are 197 city-halls that do not have the "TOTAL" row, implying that these city-halls contain a single place. Also, on average every city-hall is formed from $12,69$ places, and every province is formed from an average of $190,47$ cities. Besides the information illustrated in Table 5.1, we can extract

Category	Number	No of lines
Provinces	49	-
City-halls	9 333	10 101
Places	118 455	118 453
City-hall "TOTAL" rows	9 136	9 136
Total lines	-	137 690

Table 5.1: Characteristics of the ESPAREL dataset, corresponding to the relevant information, with the number of lines that have been extracted.

some interesting statistics about the dataset Figure 5.1 illustrates the distribution of the cities by provinces.

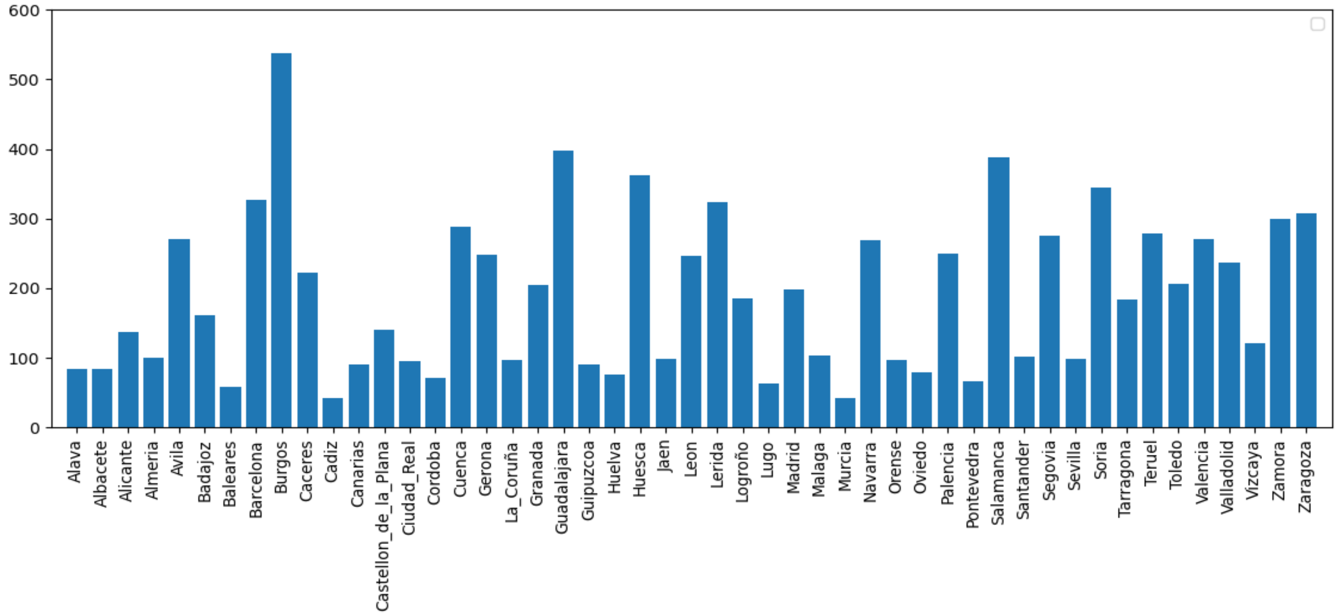


Figure 5.1: Number of cities governed by each of the 49 provinces.

Instead of plotting the number of cities by provinces, by plotting the number of places by provinces, shown in Figure 5.2, we can get a better grasp of the administrative load of each province.

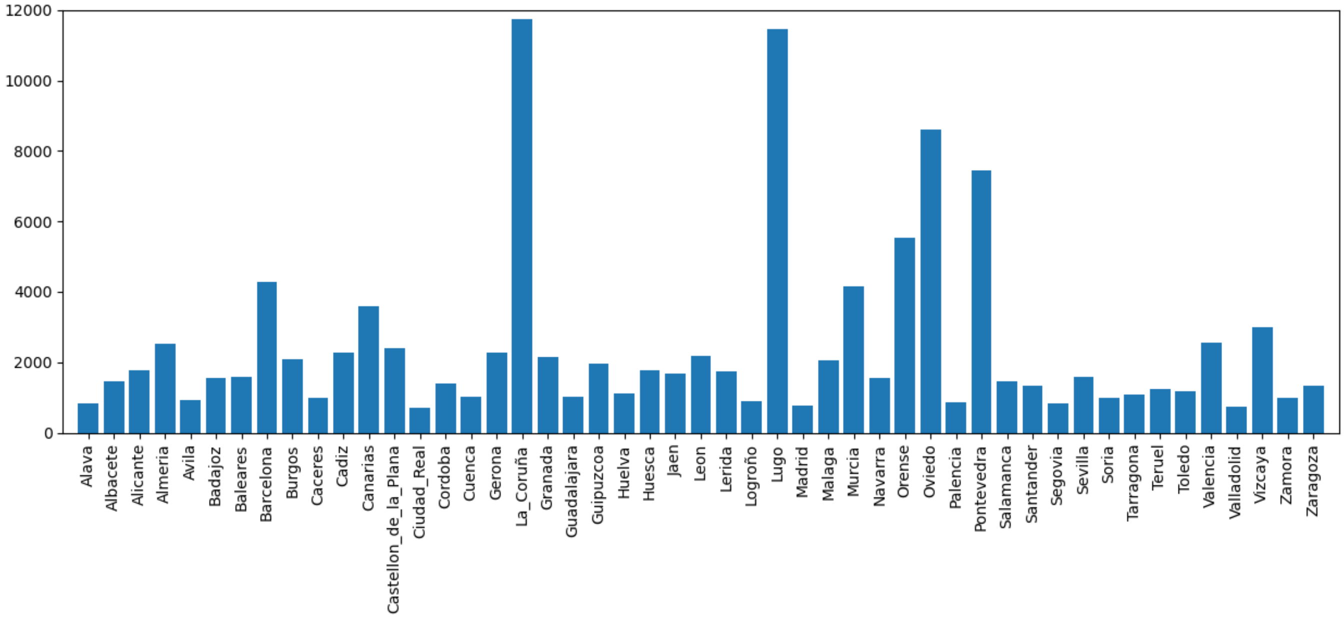


Figure 5.2: Number of places governed by each of the 49 provinces.

While many studies can be done just by analyzing the statistics of the extracted lines and the hierarchical structure of the data, the goal of this project is the transcription of the lines detailed in Table 5.1 in order to make a database to facilitate the research on this corpus. These lines and the index lines, that serve as support and are shown in Fig. 3.2 right, were extracted as explained in the previous chapter. Table 5.2 shows the total number of lines extracted of each type, which will be used in the transcription process. The relevant information about the population necessary for the ESPAREL project was included in the "Place" lines, while the relation between places and cities was included in the "Index" lines.

No. of Place lines	137 690
No. of Index lines	110 294
Total	247 984

Table 5.2: Number of lines used in the experiments.

5.2 Designing and Training the OCR System

Text recognition systems have the task of transcribing handwritten text or, in our case, printed text from images. Given that in text recognition sequences of vectors (images) are converted to text, this task can be approached from the same perspective as Automatic Speech Recognition (ASR). In the field of ASR, the fundamental problem of speech recognition consists of finding a sequence of words $\hat{W} = \hat{w}_1\hat{w}_2\ldots\hat{w}_m$ pertaining to a vocabulary $V = \{v_1, v_2, \ldots, v_N\}$, given a set of acoustic observations $O = o_1o_2\ldots o_n$. This problem can be approached by the following manner:

$$\begin{aligned}\hat{W} &= \arg\max_{W} \frac{P(O|W)P(W)}{P(O)} \\ &= \arg\max_{W} P(O|W)P(W) \qquad (5.1)\end{aligned}$$

In 5.1, $P(O)$ is the problem of preprocesing and parameterization of the signal, which can be obviated due to the fact the $P(O)$ plays no role in the maximization.

Also, in this expression $P(O|W)$ represents the acoustic model and $P(W)$ is the language model. This same approach can be used for text recognition, where the acoustic model is changed for an optical model in which instead of working with sequences of observations of cepstral coefficient vectors, the optical model works with sequences of pixels that conform the images of the lines to be transcribed. The combination of the optical model together with the language model form the OCR system that needs to be built.

As explained in Chapter 2, a widely used approach to unconstrained text recognition is the use of HMM based systems, for both optical and language modelling. However, with the arrival of neural networks and deep learning, HMM based systems fell into disuse in favor to much more expressive neural models. For this reason, in this project we used an OCR system that is based on the current technology of convolutional neural networks in combination with several layers of bi-directional long-short term memory networks (CRNN), similar to the one presented in [21].

The CRNN architecture consist of four stacked CNN layers followed by three recurrent BLSTM layers. The CNN layers form the feature extraction component of the model, and are used for the encoding of the extracted line images. These four layers consist of 12, 24, 48 and 96 filters respectively, all with convolutional kernels of 3x3 pixels. As for the activation function, for every CNN layer we used Leaky ReLU, which as demonstrated in [24] improves the results of Neural Network Acoustic Models. The convolutional layers are then passed to a Max Pooling layer of 2x2 pixels, with the exception of the third layer that is directly connected to the fourth. By reducing the size of the images with the Max Pooling layer, the CNN is capable of learning higher level features while also being more robust to the positional variance of the characters to be recognized. The summary of the CNN architecture can be seen in Table 5.3.

The second component (RNN) of the optical model is comprised of three BLSTM layers that process the output of the CNN component in left-to-right and right-to-left order, consisting of 256 units in each of the directions. A dropout of 0.5 is introduced in order to improve generalization. While the CNN component was used to extract feature from the images, the RNN component has the role

Configuration	L1	L2	L3	L4
Filters	12	24	48	96
Kernel	3x3	3x3	3x3	3x3
Activation	LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
MaxPool	2x2	2x2	-	2x2

Table 5.3: Characteristics of the CNN feature extraction component formed by 4 convolutional blocks.

of character recognition and classification based on the extracted features, taking into account both the previous and future context. By employing LSTMs, this context is memorized or forgotten depending on the importance of the context for recognizing the characters in the text. In order to be able to classify, the RNN component incorporates a linear layer, with a softmax activation function, that maps the output of the BLSTMs to an output label with a dimension of the number of characters to be recognized (70 in total) plus one for the blank symbol used by the Connectionist Temporal Classification (CTC). The purpose of CTC is to provide a probability distribution over the character transcriptions without having to align the input time steps with their corresponding output transcription, as mentioned in chapter 2. This is done by marginalizing over all the possible alignments in order to maximize the total probability of the label sequence [25].

One important aspect of this project, is that we did not use a language model either any kind of vocabulary for the recognition process. The former was not considered useful given that all words were proper names, while the latter was not available or updated for that period. Having said that, the language model distribution is set as uniform with $P(W)$ equiprobable and constant, thus making expression 5.1 and the OCR system be dependent of only the optical model.

With the architecture of the OCR system defined we proceeded to the training phase, which was carried out with the PyLaia toolkit [26]. The PyLaia toolkit, which was created for handwritten document analysis, provides an easy way for defining and training deep learning models through the command line interface. The optical model was trained from scratch by manually annotating 1 000 line images randomly selected. We guaranteed that all characters were included among these training images but diacritics were not considered necessary for this research.

From these line images, 800 were used for training and 200 for validation. The system was trained on batches of 16 line images with a learning rate of 0.0003 in order to minimize the CTC loss. Data augmentation was also used in order to artificially create more training samples through rotations, translations, scaling and shearing of the line images, with the purpose of reducing overfitting. The training ended once the character error rate (CER) on the validation set did not improve after 20 consecutive epochs. After training the system, the following 246 984 lines were recognized.

Keeping in mind that only a small percentage of the extracted lines had been annotated, the 1000 line images have been shuffled and split into training (80%) and validation (20%) sets 40 times. With this approach, all the annotated data had been used to train 40 optical models and obtain 40 transcript hypotheses for each of the lines left to be recognized. The images where then sorted based on the repetition of the most frequent transcript, with those images for which the transcript was the same or nearly the same for all the models being incorporated to the training and validation data. Note that this is an easy way for retaining the best confident transcripts.

By systematically repeating this process of training the 40 models, decoding the rest of the extracted lines and adding those lines with a perfect or nearly perfect transcript frequency to the training sets, the shortage of annotated data was greatly diminished. This process concluded once 18 120 training samples have been obtained and the increase in the accuracy of the model by adding more training data was negligible when compared to the required computational cost. With this approach, the final OCR system was much more robust than the one initially trained with just 1 000 samples.

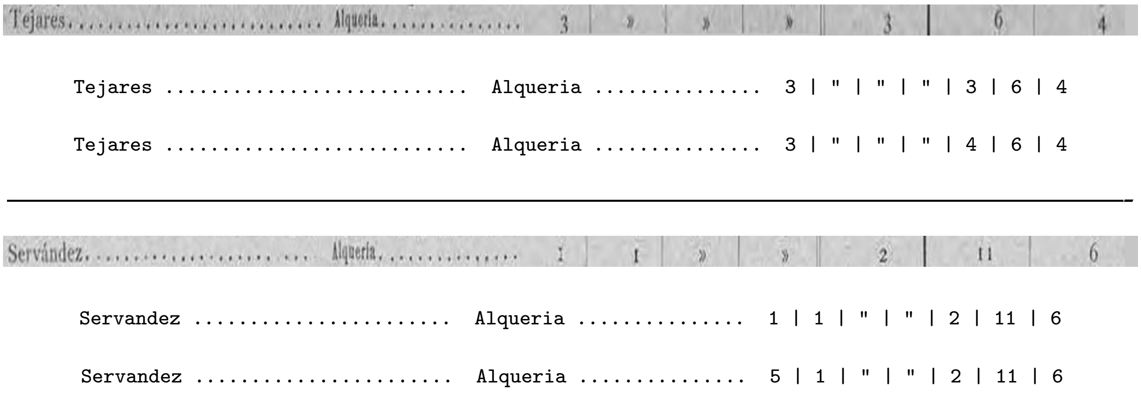
Note that although a language model was not used in these experiments, the BLSTM layers were able to capture some relations in sequences of characters and therefore the absence of a language model is alleviated. In addition, the fact that the ESPAREL corpus contained information that was replicated throughout the documents, greatly helped to detect and correct recognition errors. This is a feature that should be explored when working with documents such as the one presented in this paper.

CHAPTER 6

Results

The previous training strategy was simple and very effective and very low character errors were obtained. The errors were mainly due to noise in the images or blurred images. The character error rate was 0.05% and these errors were easily detected by using redundant information contained in throughout the corpus. That is, the use of the Index tables to detect inconsistencies in the transcriptions and to disambiguate between places with the same name belonging to different cities. In addition, the numerical data such as the number of buildings and the population living or registered could be doubled checked by contrasting this data with the information from the "TOTAL" rows, featured at the end of every city-hall. All this information coupled with the confidence score of the various transcription of the same lines gave us an easy way to detect errors that were then manually corrected.

Fig. 6.1 shows two lines that were obtained after the transcription process described in the previous chapter. In this example, there are two samples where the first line corresponds to the image of the line extracted, followed by the corresponding correct transcript, and the transcript that was obtained after the recognition process is concluded. Both lines included an error in one of the numbers. In the first example, 3 was confused with 4, and in the second example 1 was confused with 5. But as noted previously these errors could be easily detected thanks to the redundant information included in the documents. Following this approach, every line that was extracted was transcribed and the errors detected and corrected. The final results were then compiled, following the hierarchical



Tejares.......................... Alquería.............. 3 | » | » | » | 3 | 6 | 4

```
Tejares ..........................  Alqueria ..............  3 | " | " | " | 3 | 6 | 4

Tejares ..........................  Alqueria ..............  3 | " | " | " | 4 | 6 | 4
```

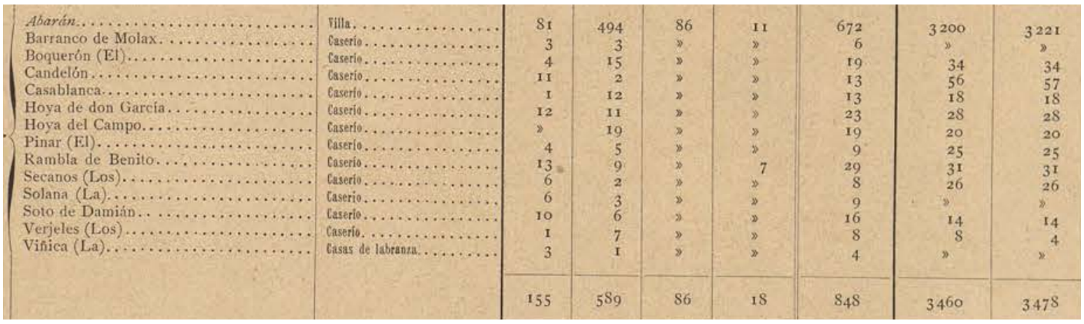
Servández...................... Alquería.............. 1 | 1 | » | » | 2 | 11 | 6

```
Servandez ......................  Alqueria ..............  1 | 1 | " | " | 2 | 11 | 6

Servandez ......................  Alqueria ..............  5 | 1 | " | " | 2 | 11 | 6
```

Figure 6.1: Example of two lines, the corresponding correct transcript, and the transcript that was obtained for each lines after the recognition process.

structure of the data. Fig. 6.2 shows an example of the image of a town and the obtained results that were moved to a spreadsheet.



Abarán..............................	Villa...............	81	494	86	11	672	3200	3221
Barranco de Molax..................	Caserío...............	3	3	»	»	6	»	»
Boquerón (El).........................	Caserío...............	4	15	»	»	19	34	34
Candelón..............................	Caserío...............	11	2	»	»	13	56	57
Casablanca............................	Caserío...............	1	12	»	»	13	18	18
Hoya de don García................	Caserío...............	12	11	»	»	23	28	28
Hoya del Campo.....................	Caserío...............	»	19	»	»	19	20	20
Pinar (El)...............................	Caserío...............	4	5	»	»	9	25	25
Rambla de Benito....................	Caserío...............	13	9	»	7	29	31	31
Secanos (Los).........................	Caserío...............	6	2	»	»	8	26	26
Solana (La)............................	Caserío...............	6	3	»	»	9	»	»
Soto de Damián......................	Caserío...............	10	6	»	»	16	14	14
Verjeles (Los).........................	Caserío...............	1	7	»	»	8	8	4
Viñica (La).............................	Casas de labranza.........	3	1	»	»	4	»	»
		155	589	86	18	848	3460	3478

81	494	86	11	672	3200	3221	Abaran	Villa
3	3	0	0	6	0	0	Barranco de Molax	Caserio
4	15	0	0	19	34	34	Boqueron El	Caserio
11	2	0	0	13	56	57	Candelon	Caserio
1	12	0	0	13	18	18	Casablanca	Caserio
12	11	0	0	23	28	28	Hoya de don Garcia	Caserio
0	19	0	0	19	20	20	Hoya del Campo	Caserio
4	5	0	0	9	25	25	Pinar El	Caserio
132	9	0	7	29	31	31	Rambla de Benito	Caserio
6	2	0	0	8	26	26	Secanos Los	Caserio
6	3	0	0	9	0	0	Solana La	Caserio
10	6	0	0	16	14	14	Soto de Damian	Caserio
1	7	0	0	8	8	4	Verjeles Los	Caserio
3	1	0	0	4	0	0	Viñica La	Casa de labranza
155	589	86	18	848	3460	3478	Total	

Figure 6.2: Image with the information of the Abaran town and the data extracted and moved to the spreadsheet.

With the information extracted and transcribed, we could calculate some very interesting statistics that highlight the value of the documents in the ESPAREL collection for conducting demographic studies. A very interesting finding is that in 1887, Spain had a population of 17 565 632 people actively living in Spanish territories, compared to 47 431 256 people according to the census of 2020 [1]. Figure 6.3 shows the distribution of the population by province, as an example of one of the many invaluable pieces of information that can be extracted from this corpus.

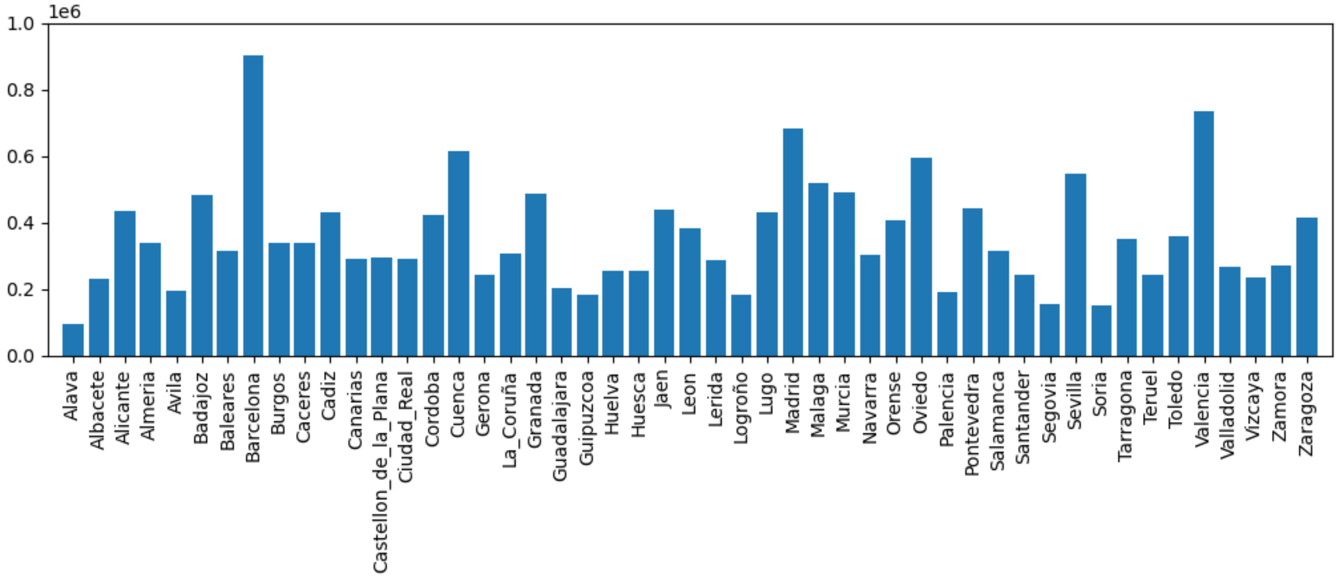


Figure 6.3: Distribution of the population living in each of the 49 provinces (“DE HECHO").

[1] https://en.wikipedia.org/wiki/Demographics_of_Spain

CHAPTER 7

Demonstrator

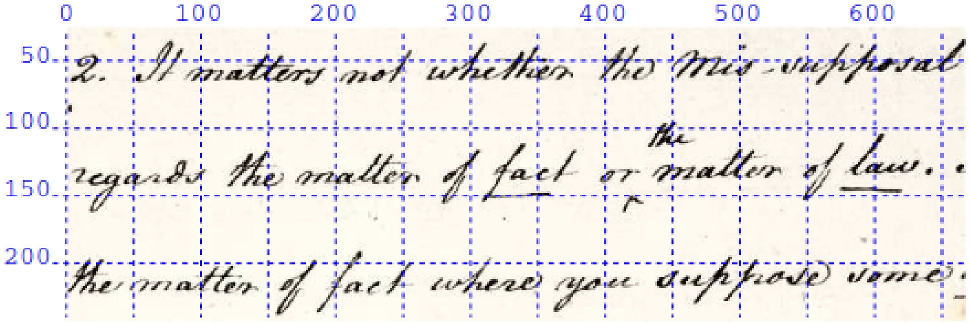
The automatic transcription of historical documents, while being a very sought after result of projects such as the one presented in this paper, a search engine with queries more complex than simple words or character sequences would further facilitate the research on this corpus. With this in mind, this chapter introduces the theoretical framework and the steps needed for building a demonstrator that captures the potential of such a search engine. This chapter will conclude with the description of the future work required for completing the aforementioned search engine, which unfortunately due to the limited time available for carrying out this project could not be included in its entirety in this paper.

7.1 Theoretical Framework

The purpose of a search engine is the retrieval of information, either through textual (query-by-string) or image based queries (query-by-example), from a collection of documents. In order to implement an Information Retrieval system the contents of the documents have to have been previously indexed. This is usually done in an offline phase that otherwise would not be practical due to the required computational cost, which guides the process of retrieving the searched information that is done in real time in an online phase. One such indexing and searching technique is known as Key Word Spotting (KWS), technique that initially had been researched in the field of speech processing [27], and which was later adopted in the field of handwritten text recognition (HTR) [28], [29].

In HTR and the automatic recognition of typed document images, the aim of KWS is to to locate all the positions of a certain keyword in a document. This technique has been developed as an alternative to the complete transcription of documents, where transcription inaccuracies would affect the retrieval of information. KWS can be used to annotate each image with word information that are likely to appear in it, along with the corresponding probabilities and corresponding positions. This technology, which has been developed by the PRHLT research center, with information generated for each image in the form of a heat map where each pixel indicates the probability that it belongs to one or more word hypotheses, is known as Probabilistic Indices (PI) [4].

As mentioned in chapter 2, the advantage of using PI is that it stores information about the n-best possible interpretations of the words to be recognized. For words that are correctly transcribed, the number of hypothesis indexed is very low, while in deteriorated regions of documents, this number is much higher. With this approach, transcription inaccuracies caused by the automatic recognition, are effectively mitigated by avoiding the loss of possible interpretations. Figure 7.1, shows an example of the probabilistic KWS index that we aim to build in this project.


0
100
200
300
400
500
600
50
100
150
200
2. It matters not whether the mis-supposal
the
regards the matter of fact or matter of law.
the matter of fact where you suppose some-

Keyword	Prob.	Bounding box
2	0.93	1 – 36 – 20 – 31
21	0.07	1 – 36 – 24 – 31
It	0.98	33 – 36 – 27 – 31
If	0.01	33 – 36 – 26 – 31
	. . .	
some	0.83	570 – 198 – 78 – 31
soner	0.02	576 – 198 – 83 – 31

Figure 7.1: An example that illustrates the information that forms a probabilistic index, source [2].

Even though PI methods are layout-agnostic, by previously having extracted the lines from the documents the indexing process can be considerably accelerated [4]. In order to generate PIs, as the one shown in Figure 7.1, we need to model the posterior distribution $P(W|O)$ where W are the transcripts given the extracted lines O. This distribution can be efficiently modelled through the use of weighted directed acyclic graphs as detailed in [2]. These graphs, known as lattices, store the n-best transcript hypotheses of the lines as shown in Figure 7.2.

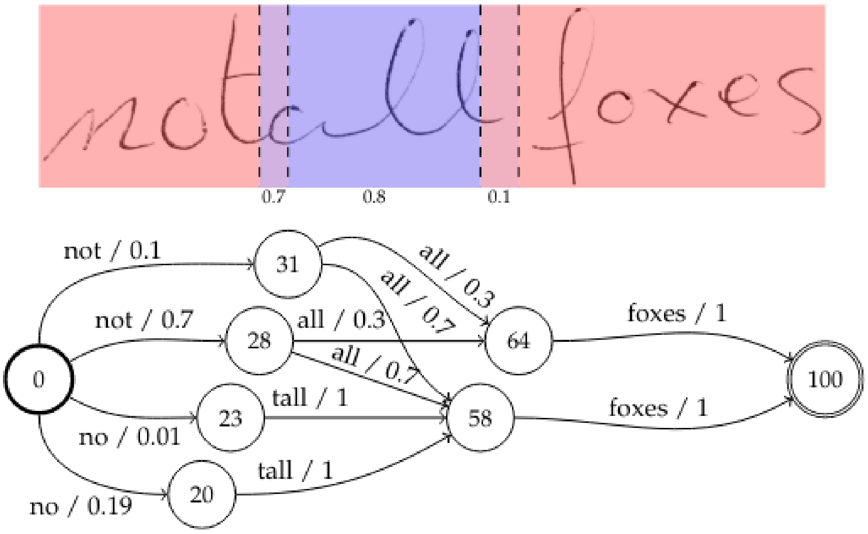


Figure 7.2: Lattice with the transcript hypotheses of an ambiguously written example phrase, source [2].

In Figure 7.2, it can be seen that for every column alignment (frame), the lattice graph models the posterior distribution of the words being explained by that frame. It can also be seen that the nodes of the graph effectively represent the segmentation of the transcription into words and the edges represent the probability of observing the word in the corresponding sequence of frames. With this information, the extraction of PI is straight forward with algorithms that solve this problem introduced in [2].

However, this still leaves the problem of building the lattices and estimating the posterior probability distribution for every frame of the extracted lines. This is done by combining the output of the optical model with a character language model, as explained hereafter.

The CRNN based optical model employed for transcribing the ESPAREL dataset, directly models the posterior distribution of the transcription hypotheses by processing a given a line image as a sequence of frames ($P(W|O)$). This implies that the neural network is capable of learning an implicit language model through the use of contextual information taken into account by the BLSTM layers. However, the keyword spotting results can be improved by incorporating an additional language model, explicitly trained from external data or from the already transcribed data. Furthermore, by training the neural network to minimize the CTC loss, the alignment of the characters/words with respect to the line image is obtained.

In order to combine the output of the optical model with the language model (typically a n-gram model), they both are then converted and represented as Weighted Finite State Transducers (WFST) which can then be combined through composition operations as detailed in [2]. From this composition, lattices as the one shown in Figure 7.2 are obtained by decoding with Viterbi algorithm with a beam search that effectively limits the size of the generated lattices. In the last step, the PI are extracted from the generated lattices, that contain the n-best sequence of characters that were decoded. The ideal output of the process is obtaining real words with a high probability assigned. However, errors in transcription frequently result in pseudo-words. These can either be these can either be strings of characters with no real meaning or words that are incorrect for the given context.

Thus, in summary, the steps needed to build the KWS based PI for indexing the ESPAREL corpus are as follows:

1. Line extraction through document layout analysis (addressed in chapter 4).
2. Design and train an optical character model (addressed in chapter 5), in order to extract the probability output for the extracted lines to be indexed (pending).
3. Train an additional language model (pending).
4. Represent and combine both the output of the optical model and the language model as a WFST (pending).

5. Prune the WFST through Viterbi decoding with a beam search in order to create the lattices (pending).

6. Extract the PI from the lattices (pending).

As it can be expected, the process of indexing a collection of documents through PI can be very computationally expensive, as each step require a combination of algorithms that have not been mentioned for the sake of simplicity. While PI are highly efficient for searching information in massive collections of documents, as demonstrated in [5], the process of extracting PI needs to be precomputed in an offline phase.

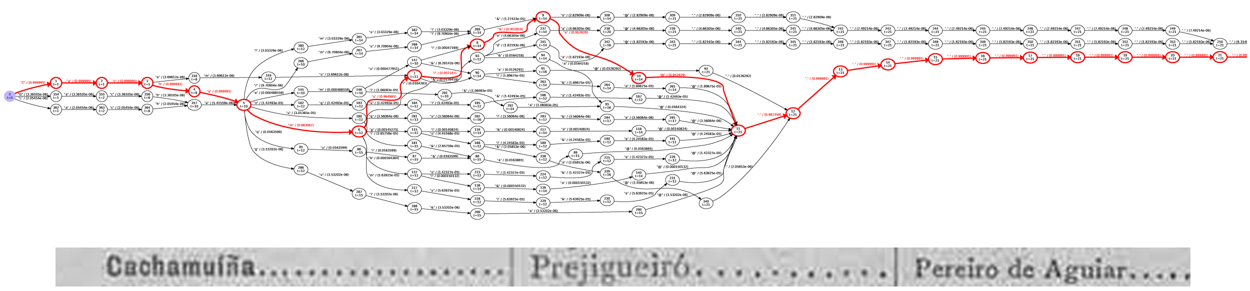
7.2 Building the Demonstrator

With the steps required for extracting PI clearly defined, in this section we'll present the implementation process for building the search engine demonstrator, with the tools needed to carry it out.

With the line extraction process and the training of the optical character model already concluded in the previous steps of the project, the next step was to obtain the posterior probability distribution of the transcription hypotheses for every extracted line. While the architecture of the model and the training of the model were carried out with the PyLaia commands `pylaia-htr-create-model` and `pylaia-htr-train-ctc`, respectively, the PyLaia toolkit offers two decoding commands for obtaining transcription hypotheses. One such command is the `pylaia-htr-decode-ctc` command, which was previously used in order to transcribe the entire ESPAREL corpus. By using this command, the toolkit enters in inference mode and provides the most probable transcription for every line of a given test collection. However, this is not sufficient for generating PI. For this purpose the PyLaia toolkit offers the `pylaia-htr-netout` command, which provides the output of the model as computed by the softmax layer, for a set of text-line images. With this, the first two steps of the process of extracting PI are now complete.

For the 3rd step we trained an order 5 n-gram character language model with the data contained in the 18 120 lines used for optimizing the optical model parameters. Next, both the output generated with the `pylaia-htr-netout` command and the language model were represented as WFST and then combined using the OpenFST library [30].

In the next step, the character lattices were obtained by decoding with Viterbi with a beam search dimension of 20. This was done with the `latgen-faster-mapped` library of the Kaldi toolkit [31]. A part of one such lattice for a given extracted line can be seen in Figure 7.3. The shape of the lattice helps visualize which regions of the lines are more difficult to transcribe as they show an accumulation of transcript hypotheses. In Figure 7.3, the path shown in red, that even though is not shown in its entirety, corresponds to the best transcription obtained by the model which also perfectly matches the reference transcription, shown last in the figure. As mentioned before, an advantage of using multiple transcript hypotheses is the fact that plausible transcriptions are not lost, and in the case of noise ridden images, the correct transcript can be indexed even though its probability is not the highest.


Cachamuiña.................... | Prejigueiró. .... | Pereiro de Aguiar.....

Cachamui&a@ ....@ | @Prejigueiro@ ....@ | @Pereiro@de@Aguiar@.....

Figure 7.3: An example of the first part of a lattice containing the transcription hypotheses of the line shown immediately below, up until the first column separator. The entire lattice can be seen in greater detail in Appendix B.

Finally, in the last step, from lattices like the one shown in Figure 7.3, the PI are then extracted with information about the n-best (20 best) most probable pseudo-words that appear in each line, together with the corresponding probability and position. This probability can then be used in order to rank the information retrieved by the PI, in order to limit the amount of information served to the user.

With the PI extracted, the demonstrator was then built following a client-server architecture that has been previously developed and used for other projects within the PRHLT investigation center. In order to make the retrieval of information more efficient, the PI have been stored in a hierarchical manner. A series of hash tables guide the searching process by first retrieving the provinces in which the terms of the queries appear, which are then refined further by retrieving the relevant document images. In addition, the hierarchical structure of the data allows the user to limit the searching space by selecting certain collections of images which could be of more interest to him/her.

In this system, the PI information generated from the lattices are linked with the coordinates of the lines extracted from the document images. This way, the pseudo-words alignments are then mapped to absolute coordinates in the image coordinate system. Note that the PI, through the probability of each pseudo-word contained, can provide the relevance score for a given word. This can be used to filter and thus reduce the amount of information supplied to the user.

Due to the complexity of the algorithms involved in extracting the PI needed for building this demonstrator, many of the implementation details have been omitted. These algorithms have been developed by the PRHLT investigation center for many years, and going into the implementation details of these algorithms would require a paper of its own. However, many of these algorithms and the steps needed for building PI are thoroughly explained in [2]. The aforementioned theses paper has been used as a guide in building the demonstrator presented in this paper.

The search engine demonstrator is accessible through a web interface [1] that allows the user to type the query in a search bar. These queries allow the user to search for individual terms, construct more complex queries through the use of the Boolean operations AND, OR and NOT (represented by the double symbols '&&', '||' and symbol '-'), or through the use of regular expressions.

[1] http://prhlt-carabela.prhlt.upv.es/esparel/

For example, the user could type in the query "Maturana" which points to the image that contains a place named Maturana pertaining to the Barrundia cityhall of Alava province (example taken from Figure 4.5 and 4.6). In order to be sure the user receives information relevant to this place, he/she could type the query "Maturana && Alava" or "Maturana && Barrundia" or even the more complex query "Maturana && (Alava || Barrundia)".

By including multiple terms into the queries without including any Boolean operators in between, the system automatically performs the AND operation of the results provided by each term individually. Thus, the previous query could also be typed "Maturana Alava". However, by combining multiple operators or through parenthesised queries the user has access to much richer and more relevant results that can make researching on the corpus easier.

7.3 Future work

The implemented search engine demonstrator illustrates the research potential of having a search engine available, with queries more complex than searching for individual terms. This demonstrator was implemented as a proof of concept, with only parts of the corpus being made available for searching (7 provinces). Besides making the entirety of the corpus accessible, this demonstrator can be further enriched with more features that will aid researching on the ESPAREL corpus.

Some of these features would be to make the demonstrator sensible to the layout of the tabular data and allow searching for information on certain columns. Coupled with features that would allow the indexed information to be refined based on numerical relations (e.g. $\leq, <, >, \geq$), would allow users to form queries for searching indexed information that satisfy a certain restriction. For example, to search for towns with a *"DE HECHO"* population of minimum 2000 inhabitants. From this a case is easily envisioned in which the data indexed in the IPs are extracted in an Excel sheet for further processing.

These kind of features would aid extract statistical conclusions from the corpus, which would be very valuable and would greatly simplify the task of researching on this corpus.

CHAPTER 8

Conclusions

This paper has introduced the problem of the automatic recognition of a historical printed collection of the Spanish census from the XIX century. The goal of this project was to get precise information by transcribing the collection of documents. This would allow researchers to conduct demographic studies by georeferencing the cities and towns and getting information about the amount and distribution of the population of that era.

In order to achieve this objective, several challenges related to paper imperfections, hierarchical structure of the data and lack of linguistic resources had to be overcome. All these challenges have been addressed by preprocessing the data, employing computer vision techniques such as image binarization, affine and morphological transformations, line detection algorithms and projection profile techniques, among others.

Firstly, the corpus of documents had to be divided by identifying the data that was necessary for building the ESPAREL dataset, the redundant data that served as support, and the data that was of no use for this project which was then removed.

Secondly, these computer vision techniques were used to mitigate the effects of the color degradation and lighting variation that affected the documents, and correct skew deviations of the pages caused by the scanning process. This helped normalize the process of document layout analysis which included the line extraction process.

Thirdly, the hierarchical structure of the data was addressed, considering that it was critical for the correct construction of the ESPAREL dataset. The images were manually organized by provinces, and then by systematically applying projection profile techniques the layout of the documents was detected. With this approach, the tabular data was segmented into subregions in order to preserve the hierarchical levels of the data. The lines of these subregions were then extracted by dividing the data into several columns, thus effectively overcoming the problem of character clipping due to the misalignment of characters.

Fourthly, out of the extracted lines 1000 were manually transcribed. These 1000 lines were then shuffled and used repeatedly for training multiple OCR models in a cascade manner, in which new lines were transcribed and included for training ensuing models. This training strategy proved to be very successful, obtaining very low character error rate without needing a large amount of annotated data. For the training of the OCR system no language model was used as we considered it would not contribute to improving the results. That being said, the use of current technology for text recognition based on CNN and BLSTM has demonstrated to be sufficiently precise to get excellent results. The redundant information that is included in these documents allowed us to easily detect and correct the few errors obtained in the recognition process. As demonstrated in this paper, this approach can be extended to deal with collections in which restrictions are present in the documents, which can be used to detect potential errors.

Finally, in order to facilitate the research on the ESPAREL corpus a search engine demonstrator was built, employing probabilistic index based technology developed by the PRHLT research center. This technology has been thoroughly tested on various projects within the PRHLT community, and was demonstrated to be very efficient for the indexing and searching of information in vast collections of historical documents. In order to build a PI based demonstrator system, we made use of many components already developed for the transcription of the corpus. By implementing a system based on PI for the indexing and searching of the data contained in the ESPAREL corpus, the loss of relevant information is minimized. With this approach the potential transcript errors are mitigated by indexing more than the most probable hypothesis. With this system, researchers interested in exploring the documents of the ESPAREL corpus are provided with an easier and richer manner of extracting relevant information. In addition, in the Future Work section of chapter 7 we mentioned several improvements that will be implemented in future versions of the demonstrator, which would further aid the research on the presented corpus.

Acknowledgment

This work has been partially supported by the BBVA Fundation, as a collaboration between the PRHLT team in charge of the HisClima project and the ESPAREL project.

Bibliography

[1] J. P. Philips and N. Tabrizi, "Historical document processing: Historical document processing: A survey of techniques, tools, and trends," *CoRR*, vol. abs/2002.06300, 2020. [Online]. Available: https://arxiv.org/abs/2002.06300

[2] J. Puigcerver, "A probabilistic formulation of keyword spottin," Ph.D. dissertation, Universitat Politècnica de València, 2018.

[3] D. A. Borges Oliveira and M. P. Viana, "Fast cnn-based document layout analysis," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1173–1180.

[4] E. Vidal, V. Romero, A. H. Toselli, J. Sánchez, V. Bosch, L. Quirós, J. M. Benedí, J. R. Prieto, M. Pastor, F. Casacuberta, C. Alonso, C. García, L. Márquez, and C. Orcero, "The carabela project and manuscript collection: Large-scale probabilistic indexing and content-based classification," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 85–90.

[5] E. Vidal, "Text search and information retrieval in large historical collections of untranscribed manuscripts." Invited key note talk at International Conference on Document Analysis and Recognition (ICDAR), 2019.

[6] S. Chandna, F. Rindone, C. Dachsbacher, and R. Stotzka, "Quantitative exploration of large medieval manuscripts data for the codicological research," in *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*, 2016, pp. 20–28.

[7] A. Fischer, M. Baechler, A. Garz, M. Liwicki, and R. Ingold, “A combined system for text line extraction and handwriting recognition in historical documents,” in *2014 11th IAPR International Workshop on Document Analysis Systems*, 2014, pp. 71–75.

[8] E. Chammas, C. Mokbel, and L. Likforman-Sulem, “Handwriting recognition of historical documents with few labeled data,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 43–48.

[9] P. Le, N. Nayef, M. Visani, J.-M. Ogier, and D. Tran, “Text and non-text segmentation based on connected component features,” 08 2015.

[10] A. W. Harley, A. Ufkes, and K. G. Derpanis, “Evaluation of deep convolutional nets for document image classification and retrieval,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 991–995.

[11] L. Kang, J. Kumar, P. Ye, Y. Li, and D. Doermann, “Convolutional neural networks for document image classification,” in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 3168–3172.

[12] D. A. Borges Oliveira and M. P. Viana, “Fast cnn-based document layout analysis,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1173–1180.

[13] G. Nagy, S. Seth, and M. Viswanathan, “A prototype document image analysis system for technical journals,” *Computer*, vol. 25, pp. 10–22, 1992.

[14] W. Zhu, Q. Chen, C. Wei, and Z. Li, “A segmentation algorithm based on image projection for complex text layout,” vol. 1890, 10 2017, p. 030011.

[15] M. Mohamed and P. Gader, “Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 548–554, 1996.

[16] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Suen, “An hmm-based approach for off-line unconstrained handwritten word modeling and recogni-

tion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 752–760, 1999.

[17] M.-Y. Chen, A. Kundu, and S. Srihari, "Variable duration hidden markov model and morphological segmentation for handwritten word recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1993, pp. 600–601.

[18] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[19] C. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," *CoRR*, vol. abs/1603.03101, 2016. [Online]. Available: http://arxiv.org/abs/1603.03101

[20] T. M. Breuel, "High performance text recognition using a hybrid convolutional-lstm implementation," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 11–16.

[21] E. Lang, J. Puigcerver, A. H. Toselli, and E. Vidal, "Probabilistic indexing and search for information extraction on handwritten german parish records," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 44–49.

[22] O. Boudraa, W. Hidouci, and D. Michelucci, "Using skeleton and hough transform variant to correct skew in historical documents," *Mathematics and Computers in Simulation*, vol. 167, 06 2019.

[23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[24] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models," 2013.

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with

recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[26] J. Puigcerver and C. Mocholí, "Pylaia," https://github.com/jpuigcerver/PyLaia, 2018.

[27] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *International Conference on Acoustics, Speech, and Signal Processing,*, 1989, pp. 627–630 vol.1.

[28] S. Khoubyari and J. J. Hull, "Keyword location in noisy document image," in *In 2nd Annual Symposium on Document Analysis and Information Retrieval*, 1993, pp. 217–231.

[29] F. Chen, L. Wilcox, and D. Bloomberg, "Word spotting in scanned images using hidden markov models," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 1993, pp. 1–4 vol.5.

[30] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "Openfst: A general and efficient weighted finite-state transducer library," in *Implementation and Application of Automata*, J. Holub and J. Žd'árek, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 11–23.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

APPENDIX A

Software requirements

The software solution, for the transcription of the ESPAREL corpus, proposed in this project consists of three main components: 1) line extraction module; 2) OCR module; and 3) demonstrator module. In this appendix the software requirements will be covered, detailing all the necessary libraries and tool-kits for each one of the aforementioned components in order to allow for the replication of the experiments presented in this paper.

The **line extraction module**, with all the sub-components for skew correction, document layout analysis and image preprocessing, was entirely programmed in Python with the use of the computer vision library OpenCV [1]. Even though this component was written in Python 3.8 with OpenCV version 4.5.1, which features newer and more efficient functions, the code needed to develop the component can be written in any version of Python 3.6+ with all functions employed available in OpenCV version 3.0 or higher.

For the **OCR module**, various bash commands were used for the preparation and partitioning of the data for training and testing. The training phase was carried out with the PyLaia toolkit [26], which as defined by its authors is *"a device agnostic, PyTorch based, deep learning toolkit for handwritten document analysis"*. In order to accelerate the training phase and provide GPU support, this toolkit requires CUDA [2] and CuDNN [3]. For this project we used a certain PyLaia version, which was not affected by a bug in the extraction of the char/word segmentation

[1] http://opencv.org/

[2] https://developer.nvidia.com/cuda-toolkit

[3] https://developer.nvidia.com/cudnn

boundaries. This was important for building the search engine demonstrator detailed in Chapter 7. This PyLaia version required Python v3.6.12, Torch v1.0.1, CUDA v10.0 and CuDNN v7.6.5, among other dependencies which were usually satisfied by default or did not require such strict versions. The decoding process was done using version 5.5 of the Kaldi Speech Recognition Toolkit [31].

For the **demonstrator module**, in addition to the PyLaia and Kaldi toolkit that have been already mentioned, we used version 1.7.3 of the SRILM toolkit for building a n-gram language model from the transcribed lines.

In summary, all the toolkits and libraries used in this project can be seen below:

- Python v3.6 and 3.8, an interpreted programming language;
- OpenCV v4.5.1, an open source computer vision library;
- SRILM v1.7.3, a toolkit for building and applying statistical language models;
- Kaldi v5.5, a toolkit for speech recognition that can also be used for handwritten text recognition tasks;
- OpenFST v1.7.2, as part of the Kaldi toolkit for building WFST.
- CUDA v10.0, a parallel computing platform and application programming interface (API);
- CuDNN v7.6.5, a GPU-accelerated library of primitives for deep neural networks;
- PyLaia git 41d2cc41d742e7ab336393fde8f56585ff49ee52, a deep learning toolkit for handwritten document analysis;
- Torch v1.0.1, an end-to-end open source platform for machine learning;
- Torchvision v0.2.2, a component of the PyTorch open source machine learning framework;
- Pillow>=5.2, an open source library that adds image processing capabilities to Python;
- scipy, an open-source software for mathematics, science, and engineering.

- editdistance, a library that implements the Levenshtein distance efficiently.

APPENDIX B

Lattice Example

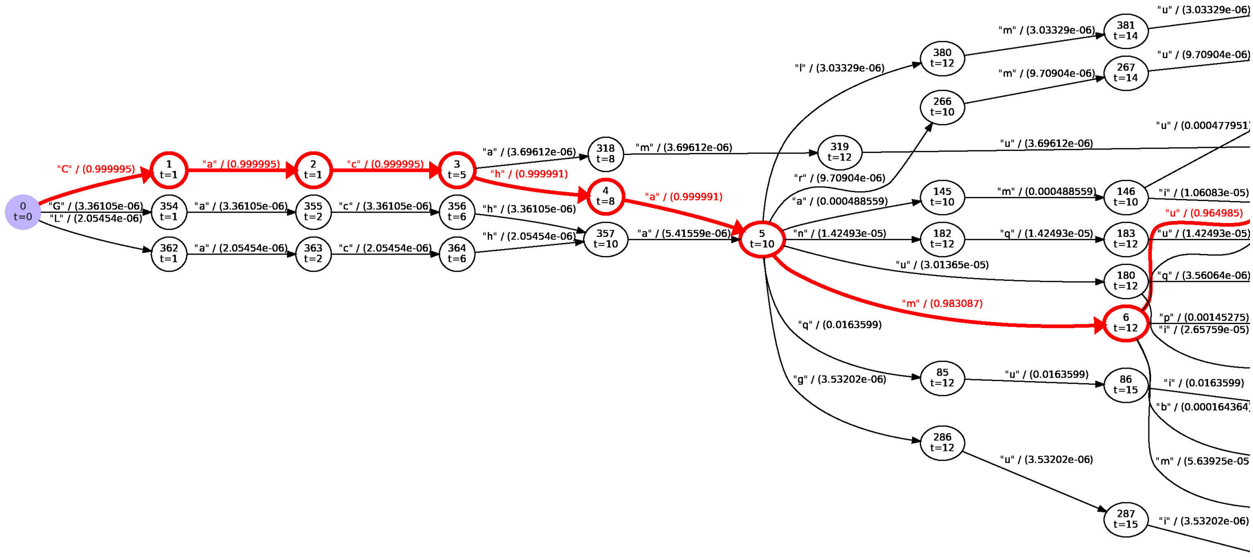


Figure B.1: Lattice example Part 1/13 .

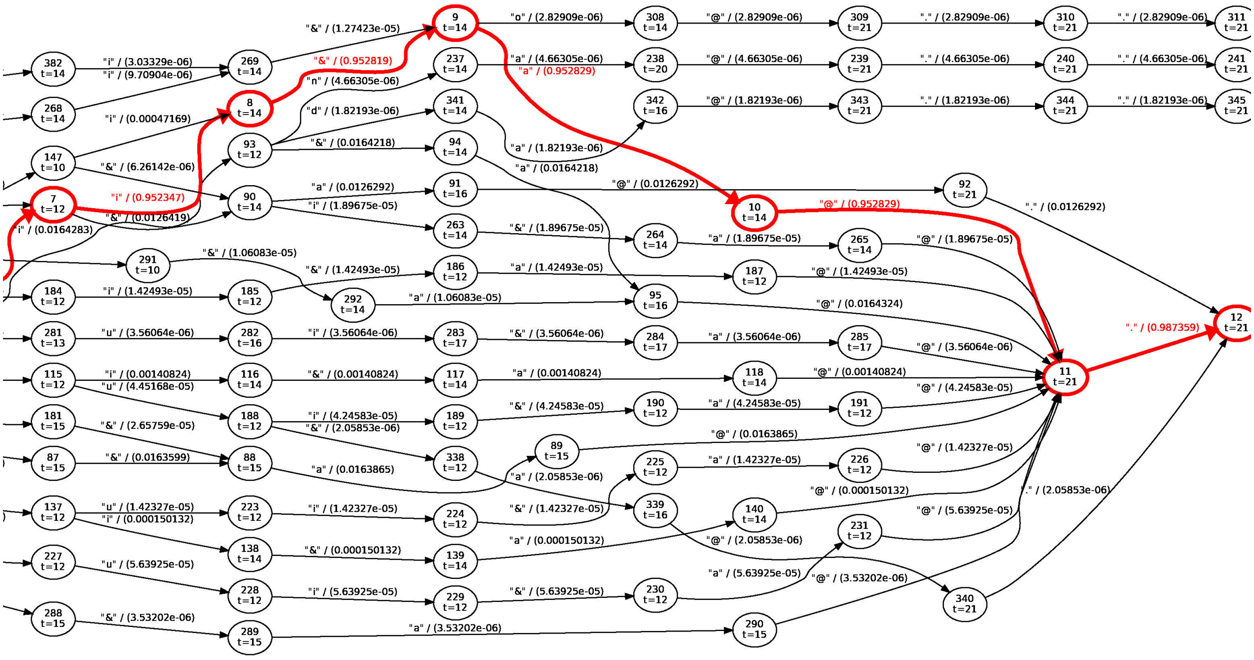


Figure B.2: Lattice example Part 2/13 .

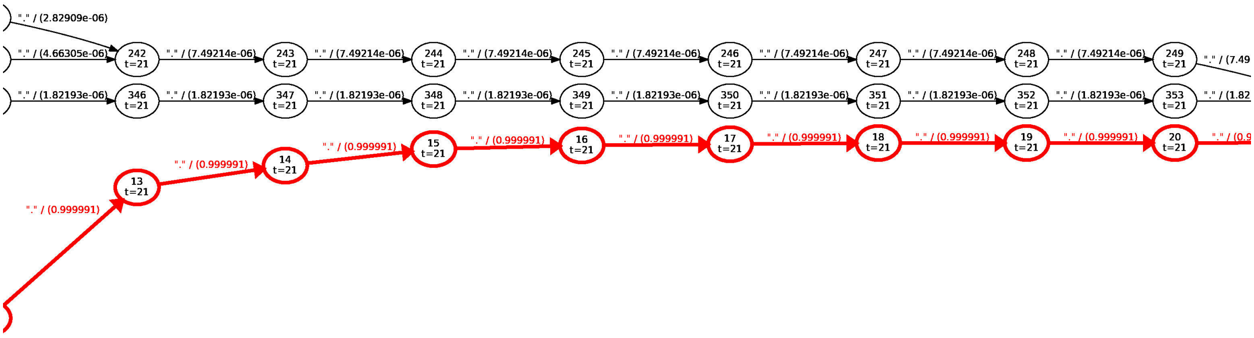


Figure B.3: Lattice example Part 3/13 .

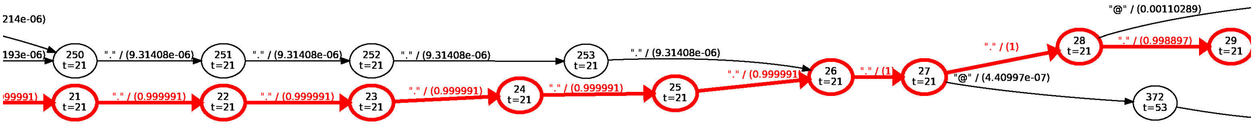


Figure B.4: Lattice example Part 4/13 .

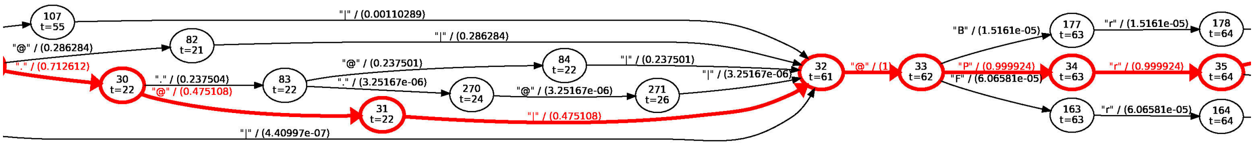


Figure B.5: Lattice example Part 5/13 .

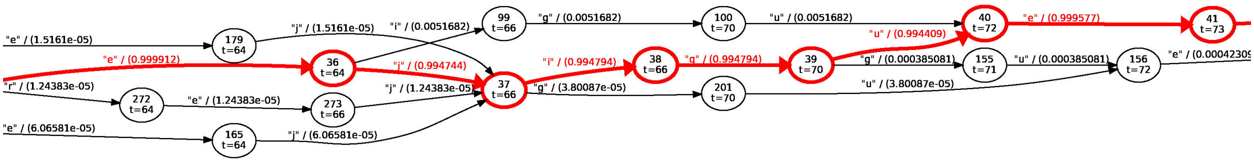


Figure B.6: Lattice example Part 6/13 .

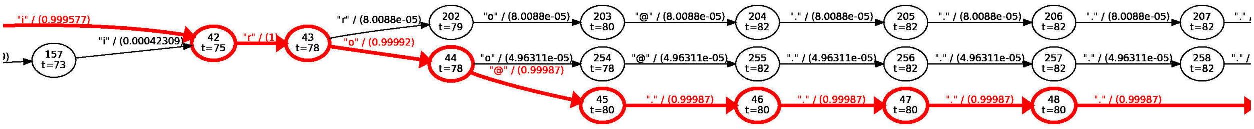


Figure B.7: Lattice example Part 7/13 .

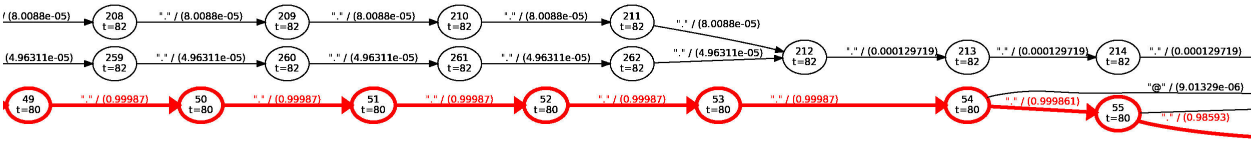


Figure B.8: Lattice example Part 8/13 .

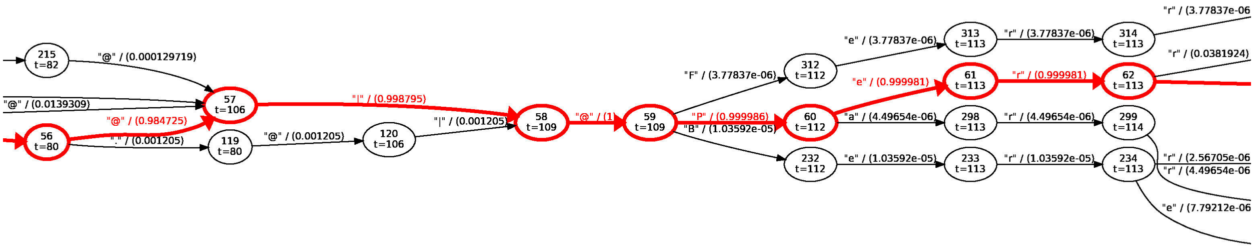


Figure B.9: Lattice example Part 9/13 .

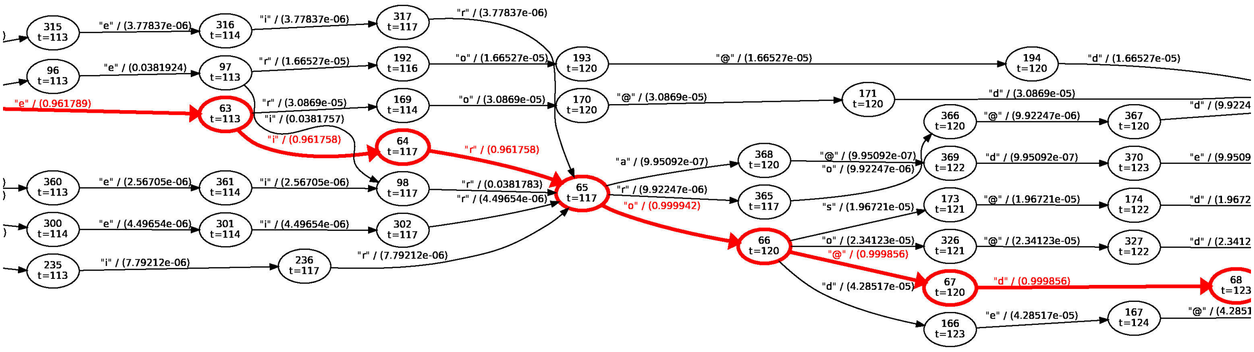


Figure B.10: Lattice example Part 10/13 .

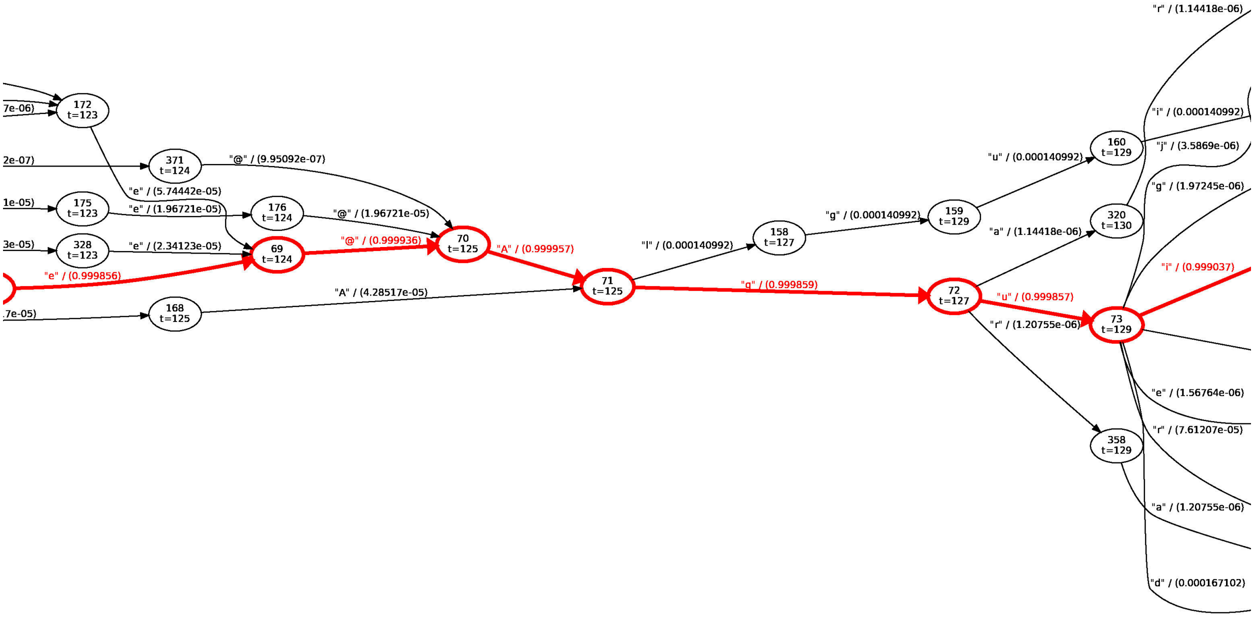


Figure B.11: Lattice example Part 11/13 .

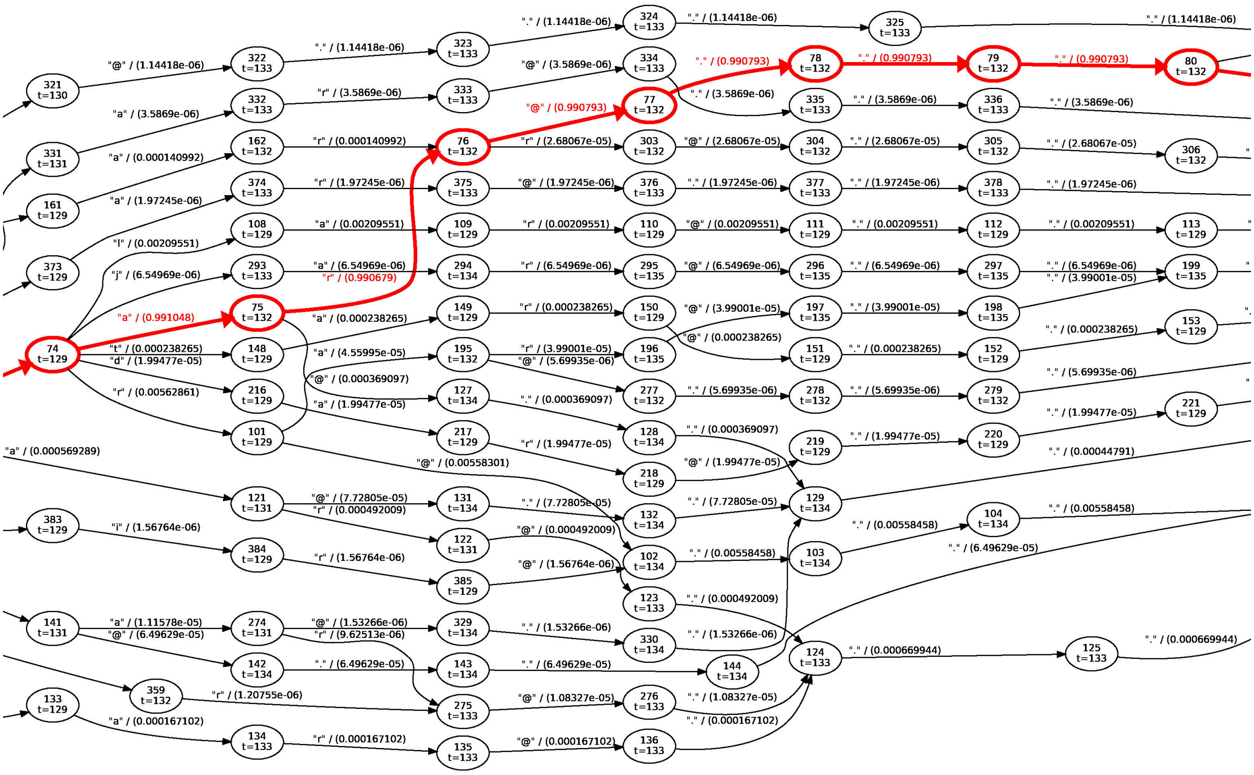


Figure B.12: Lattice example Part 12/13 .

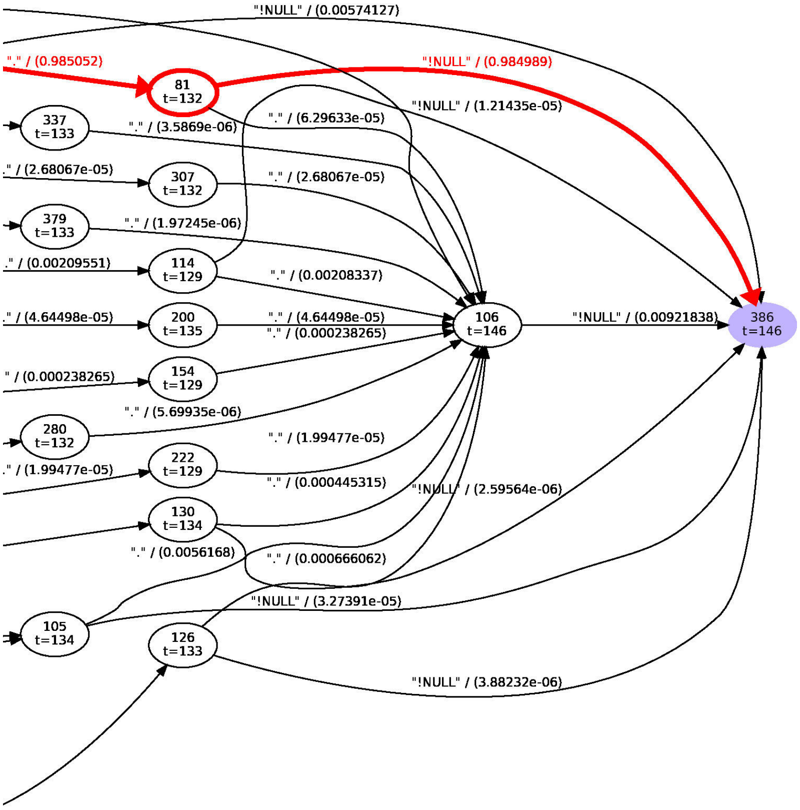


Figure B.13: Lattice example Part 13/13.