UNIVERSITAT POLITÈCNICA DE VALÈNCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

PATTERN RECOGNITION AND
HUMAN LANGUAGE TECHNOLOGY GROUP

PHD THESIS

Advances in Document Layout Analysis

Vicente Bosch Campos

Supervised by Dr. Enrique Vidal Ruiz
and Dr. Alejandro Héctor Toselli

October 21, 2019

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

ABSTRACT

Handwritten Text Segmentation (HTS) is a task within the Document Layout Analysis field that aims to detect and extract the different page regions of interest found in handwritten documents. HTS remains an active topic, that has gained importance with the years, due to the increasing demand to provide textual access to the myriads of handwritten document collections held by archives and libraries.

This thesis considers HTS as a task that must be tackled in two specialized phases: detection and extraction. We see the detection phase fundamentally as a recognition problem that yields the vertical positions of each region of interest as a by-product. The extraction phase consists in calculating the best contour coordinates of the region using the position information provided by the detection phase.

Our proposed detection approach allows us to attack both higher level regions: paragraphs, diagrams, etc., and lower level regions like text lines. In the case of text line detection we model the problem to ensure that the system's yielded vertical position approximates the fictitious line that connects the lower part of the grapheme bodies in a text line, commonly known as the baseline.

One of the main contributions of this thesis, is that the proposed modelling approach allows us to include prior information regarding the layout of the documents being processed. This is performed via a Vertical Layout Model (VLM).

We develop a Hidden Markov Model (HMM) based framework to tackle both region detection and classification as an integrated task and study the performance and ease of use of the proposed approach in many corpora. We review the modelling simplicity of our approach to process regions at different levels of information: text lines, paragraphs, titles, etc. We study the impact of adding deterministic and/or probabilistic prior information and restrictions via the VLM that our approach provides.

Having a separate phase that accurately yields the detection position (baselines in the case of text lines) of each region greatly simplifies the problem that must be tackled during the extraction phase. In this thesis we propose to use a distance map that takes into consideration the grey-scale information in the image. This allows us to yield extraction frontiers which are equidistant to the adjacent text regions. We study how our approach escalates its accuracy proportionally to the quality of the provided detection vertical position. Our extraction approach gives near perfect results when human reviewed baselines are provided.

We propose specific evaluation measures to assess the accuracy not only of region detection results, but also of our region-type classification. We also compare these measures with the classical extraction-based evaluation measures and with other more recently proposed baseline based measures. We study how these measures reflect the impact the proposed solutions have on Handwritten Text Recognition (HTR) systems that are dependent on them.

In order to make our framework practical we must ensure that it is applicable in large production scenarios. To this end we designed an iterative semi-automatic process that is able to generate groundtruth quality region detection while reducing the amount of human effort required.

We also study layout analysis problems where region detection and classification tasks are impossible to be performed using only graphical information. In many cases this happens because the considered document does not actually exhibit any graphical clue to distinguish between adjacent regions. Specifically, we consider the very frequent situation where only by actually reading the text in the document is it possible to determine which parts of the text belong to a region or to another. To tackle this situation we study the (additional) use of text content features to improve the accuracy of our HTS approach.

Since layout analysis has traditionally been considered a step that must be performed previously to any sort of text recognition, wanting to use text content based features generates a vicious circle: text recognition requires previous layout analysis to be performed, but in its turn, some sort of text recognition is required to obtain these textual content features required for the layout analysis. To circumvent this deadlock we rely on the recently introduced concept of "Probabilistic Index" which, for a given page image, provides the probability that a word appears in every word-sized bounding box of the image. We show that the textual content information features extracted from these indexes can greatly improve the classification accuracy of any HTS approach. We review how these novel text content features can be extracted for use in specific tasks and show that they do actually boost the performance of our proposed HTS approach.

Finally, we study a tandem approach to HTS that performs a simple combination of a Convolutional Neural Network (CNN) with our methods based on VLMs and HMMs. The CNN proves to be very effective at providing an accurate, pixel-wise preprocess of the input images, while the VLM and HMM models take into account contextual prior information. In conclusion, we consider that the proposed tandem system outperforms both standalone HMM and CNN systems.

RESUMEN

La Segmentación de Texto Manuscrito (STM) es una tarea dentro del campo de investigación de Análisis de Estructura de Documentos (AED) que tiene como objetivo detectar y extraer las diferentes regiones de interés de las páginas que se encuentran en documentos manuscritos. La STM es un tema de investigación activo que ha ganado importancia con los años debido a la creciente demanda de proporcionar acceso textual a las miles de colecciones de documentos manuscritos que se conservan en archivos y bibliotecas.

Esta tesis entiende la STM como una tarea que debe ser abordada en dos fases especializadas: detección y extracción. Consideramos que la fase de detección es, fundamentalmente, un problema de clasificación cuyo subproducto son las posiciones verticales de cada región de interés. Por su parte, la fase de extracción consiste en calcular las mejores coordenadas de contorno de la región utilizando la información de posición proporcionada por la fase de detección.

Nuestro enfoque de detección nos permite atacar tanto regiones de alto nivel (párrafos, diagramas...) como regiones de nivel bajo (líneas de texto principalmente). En el caso de la detección de líneas de texto, modelamos el problema para asegurar que la posición vertical estimada por el sistema se aproxime a la línea ficticia que conecta la parte inferior de los cuerpos de los grafemas en una línea de texto, comúnmente conocida como línea base.

Una de las principales aportaciones de esta tesis es que el enfoque de modelización propuesto nos permite incluir información conocida a priori sobre la disposición de los documentos que se están procesando. Esto se realiza mediante un Modelo de Estructura Vertical (MEV).

Desarrollamos un marco de trabajo basado en los Modelos Ocultos de Markov (MOM) para abordar tanto la detección de regiones como su clasificación de forma integrada, así como para estudiar el rendimiento y la facilidad de uso del enfoque propuesto en numerosos corpus. Así mismo, revisamos la simplicidad del modelado de nuestro enfoque para procesar regiones en diferentes niveles de información: líneas de texto, párrafos, títulos, etc. Finalmente, estudiamos el impacto de añadir información y restricciones previas deterministas o probabilistas a través de el MEV propuesto que nuestro enfoque proporciona.

Disponer de un método independiente que obtiene con precisión la posición de cada región detectada (líneas base en el caso de las líneas de texto) simplifica enormemente el problema que debe abordarse durante la fase de extracción. En esta tesis proponemos utilizar un mapa de distancias que tiene en cuenta la información de escala de grises de la imagen. Esto nos permite obtener fronteras de extracción que son equidistantes a las regiones de texto adyacentes. Estudiamos como nuestro enfoque aumenta su precisión de manera proporcional a la calidad de la detecttion y descubrimos que da resultados casi perfectos cuando se le proporcionan líneas de base revisadas por humanos.

Proponemos medidas de evaluación específicas para evaluar la precisión tanto de los resultados de detección de regiones como de su clasificación. También comparamos estas medidas con las

medidas clásicas de evaluación basadas en la extracción y con otras medidas recientes que se basan en las líneas base para realizar su evaluación. Estudiamos cómo estas medidas reflejan el impacto que tienen las soluciones propuestas tienen en los sistemas de Reconocimiento de Texto Manuscrito (RTM) que dependen de ellas.

Para que nuestro método resulte práctico, debemos asegurarnos de que sea aplicable en escenarios de producción con grandes cantidades de datos. Para ello, hemos diseñado un proceso semiautomático iterativo que es capaz de detectar las regiones con precisión similar a la de un humano. Esto reduce, a su vez, la cantidad de esfuerzo humano requerido.

También estudiamos los corpus problemáticos en los que las tareas de detección y clasificación de regiones son imposibles de realizar utilizando solo información gráfica. En muchos casos esto sucede porque el documento que se está analizando no muestra ninguna pista gráfica para distinguir entre regiones adyacentes. Específicamente, examinamos la situación frecuente en la que solo leyendo el texto en el documento es posible determinar qué partes del texto pertenecen a una región o a otra. Para abordar esta situación, analizamos el uso (adicional) de características que se calculan en base al contenido textual del documento para mejorar la precisión de nuestro enfoque RTM.

Dado que el análisis de estructura de documentos se considera tradicionalmente un paso que debe realizarse antes de cualquier tipo de reconocimiento de texto, el hecho de querer utilizar características basadas en el contenido textual genera un círculo vicioso: el reconocimiento de texto necesita que se realice un análisis de estructura previo, pero, a su vez, se requiere algún tipo de reconocimiento de texto para obtener estas características, basadas en el contenido del texto, necesarias para el análisis de la estructura del documento. Para evitar esta encrucijada, nos valemos del concepto recientemente introducido del "índice probabilístico", que, para una imagen de página dada, ofrece la probabilidad de que una palabra aparezca en cada recuadro de la imagen del tamaño de una palabra. Demostramos que las características basadas en contenido textual, que se calculan a través de estos índices, pueden mejorar enormemente la precisión de clasificación de cualquier enfoque STM. Revisamos cómo se pueden calcular estas novedosas características para su uso en tareas específicas y demostramos que realmente mejoran el rendimiento de nuestro enfoque propuesto para el STM.

Finalmente, abordamos un enfoque tándem para la STM que realiza una simple combinación de una Red Neural Convolucional (RNC) con nuestros métodos basados en MEV y MOM. La RNC demuestra ser muy eficaz a la hora de proporcionar un preprocesamiento preciso en píxeles de las imágenes de entrada, mientras que los modelos MEV y MOM tienen en cuenta la información contextual previa. En conclusion, consideramos que el sistema tándem propuesto supera tanto a los sistemas MOM como a los sistemas RNC independientes.

RESUM

La Segmentació de Text Manuscrit (STM) és una tasca dins del camp d'investigació d'Anàlisi d'Estructura de Documents (AED) que té com a objectiu detectar i extraure les diferents regions d'interès de les pàgines que es troben en documents manuscrits. La STM és un tema d'investigació actiu que ha guanyat importància amb els anys a causa de la creixent demanda per proporcionar accés textual als milers de col·leccions de documents manuscrits que es conserven en arxius i biblioteques.

Aquesta tesi entén la STM com una tasca que ha de ser abordada en dues fases especialitzades: detecció i extracció. Considerem que la fase de detecció és, fonamentalment, un problema de classificació el subproducte de la qual són les posicions verticals de cada regió d'interès. Per la seva part, la fase d'extracció consisteix a calcular les millors coordenades de contorn de la regió utilitzant la informació de posició proporcionada per la fase de detecció.

El nostre enfocament de detecció ens permet atacar tant regions d'alt nivell (paràgrafs, diagrames ...) com regions de nivell baix (línies de text principalment). En el cas de la detecció de línies de text, modelem el problema per a assegurar que la posició vertical estimada pel sistema s'aproximi a la línia fictícia que connecta la part inferior dels cossos dels grafemes en una línia de text, comunament coneguda com a línia base.

Una de les principals aportacions d'aquesta tesi és que l'enfocament de modelització proposat ens permet incloure informació coneguda a priori sobre la disposició dels documents que s'estan processant. Això es realitza mitjançant un Model d'Estructura Vertical (MEV).

Desenvolupem un marc de treball basat en els Models Ocults de Markov (MOM) per a abordar tant la detecció de regions com la seva classificació de forma integrada, així com per a estudiar el rendiment i la facilitat d'ús de l'enfocament proposat en nombrosos corpus. Així mateix, revisem la simplicitat del modelatge del nostre enfocament per a processar regions en diferents nivells d'informació: línies de text, paràgrafs, títols, etc. Finalment, estudiem l'impacte d'afegir informació i restriccions prèvies deterministes o probabilistes a través del MEV que el nostre mètode proporciona.

Disposar d'un mètode independent que obté amb precisió la posició de cada regió detectada (línies base en el cas de les línies de text) simplifica enormement el problema que ha d'abordar-se durant la fase d'extracció. En aquesta tesi proposem utilitzar un mapa de distàncies que té en compte la informació d'escala de grisos de la imatge. Això ens permet obtenir fronteres d'extracció que són equidistants de les regions de text adjacents. Estudiem com el nostre enfocament augmenta la seva precisió de manera proporcional a la qualitat de la detecció i descobrim que dona resultats quasi perfectes quan se li proporcionen línies de base revisades per humans.

Proposem mesures d'avaluació específiques per a avaluar la precisió tant dels resultats de detecció de regions com de la seva classificació. També comparem aquestes mesures amb les mesures clàssiques d'avaluació basades en l'extracció i amb altres mesures recents que es basen en les línies

base per a realitzar la seva avaluació. Estudiem com aquestes mesures reflecteixen l'impacte que tenen les solucions proposades en els sistemes de Reconeixement de Text Manuscrit (RTM) que depenen d'elles.

Perquè el nostre mètode resulti pràctic, hem d'assegurar-nos que sigui aplicable en escenaris de producció amb grans quantitats de dades. Per a això, hem dissenyat un procés semiautomàtic iteratiu que és capaç de detectar les regions amb precisió similar a la d'un humà. Això redueix, al seu torn, la quantitat d'esforç humà requerit.

També estudiem els corpus problemàtics en els quals les tasques de detecció i classificació de regions són impossibles de realitzar utilitzant només informació gràfica. En molts casos això succeeix perquè el document que s'està analitzant no mostra cap pista gràfica per a distingir entre regions adjacents. Específicament, examinem la situació freqüent en la qual només llegint el text en el document és possible determinar quines parts del text pertanyen a una regió o a una altra. Per a abordar aquesta situació, analitzem l'ús (addicional) de característiques que es calculen sobre la base del contingut textual del document per a millorar la precisió del nostre enfocament STM.

Atès que l'anàlisi d'estructura de documents es considera tradicionalment un pas que ha de realitzar-se abans de qualsevol mena de reconeixement de text, el fet de voler utilitzar característiques basades en el contingut textual genera un cercle viciós: el reconeixement de text necessita que es realitzi una anàlisi d'estructura previ, però, al seu torn, es requereix algun tipus de reconeixement de text per a obtenir aquestes característiques, basades en el contingut del text, necessàries per a l'anàlisi de l'estructura del document. Per a evitar aquesta cruïlla, ens valem del concepte recentment introduït de l'"índex probabilístic", que, per a una imatge de pàgina donada, ofereix la probabilitat que una paraula aparegui en cada requadre de la imatge de la grandària d'una paraula. Vam demostrar que les característiques basades en contingut textual, que es calculen a través d'aquests índexs, poden millorar enormement la precisió de classificació de qualsevol enfocament STM. Revisem com es poden calcular aquestes noves característiques per al seu ús en tasques específiques i vam demostrar que realment milloren el rendiment del nostre enfocament proposat per al STM.

Finalment, abordem un enfocament tàndem per a la STM que realitza una simple combinació d'una Xarxa Neural Convolucional (XNC) amb els nostres mètodes basats en MEV i MOM. La XNC demostra ser molt eficaç a l'hora de proporcionar un preprocessament precís en píxels de les imatges d'entrada, mentre que els models MEV i MOM tenen en compte la informació contextual prèvia. En conclusió, considerem que el sistema tàndem proposat supera tant els sistemes MOM com els sistemes XNC independents.

ACKNOWLEDGEMENTS

This thesis gives closure to a very important chapter in my life. When I first thought of changing my professional career to become a researcher and embark in this challenge, I was rather naive on the problems it would entail. Luckily, I have great people in my life that have helped me, in one way or another, along this voyage.

First of all, I would like to thank my thesis directors: *Enrique Vidal*, who gave me the wonderful opportunity of joining the *PRHLT* group under his tutelage and to *Alejandro H. Toselli* for his guidance and help. Thank you both for the time, dedication and predisposition to help me. This thesis is as much theirs as it is mine.

I would like to thank my colleagues (present and past) of the *PRHLT* group that have helped me tremendously during this years in the group. I would specially like to thank inside the *PRHLT* all my colleagues and seniors in the Handwritten Text Recognition section. To *Verónica Romero* for her patience and help in all of the research studies we have collaborated. To *Dani Martín-Albo* and *Paco Álvaro* my initial desk mates who lead in my first years as a junior researcher. To *Joan Puigcerver* and *Lorenzo Quirós* my current colleagues who have been with me during the last struggles of my research.

This endeavour has also been possible thanks to many people outside of the research world. Thank you to my friends in Almussafes and Benimaclet.

Finally I would like to thank my family and specially my wife Beatriz, without your patience, care and encouragement this work might have never reached completion.

Valencia, June 2019

PREFACE

The main goal of this thesis is to provide a robust approach to perform Handwritten Text Segmentation (HTS) in documents. The approach tackles HTS as a problem with a detection phase and an extraction phase.

In order to explain in detail the approach, the thesis has been organized in eight chapters that are best read in consecutive order. Chapter 1, covers the basic preliminaries of HTS, introducing the problem, commenting on some issues with current approaches, providing the theoretical framework upon which the rest of the thesis is build and listing the set of expected scientific outcomes.

In Chapter 2 we review the current state of the art of HTS solutions. We do so by providing our own modified taxonomy on the problems and approaches that have been published. We finish this chapter by classifying our detection and extraction approaches.

Chapter 3 presents our approach on text region detection and classification in earnest. The chapter starts by presenting a classical reference detection method which we will use for comparison reasons. We provide a detailed explanation of our stochastic approach to text region classification and detection, how it models the page layout, how the models are trained and used to decode new pages. This chapter finishes with three novel ideas regarding how to use this framework for real production scenarios, the tandem combination with Convolutional Neural Networks and the use of text content based features.

Next, Chapter 4 covers our extraction technique that uses the results yielded by the detection method to calculate equidistant extraction frontiers to the adjacent text regions. The method performs this by means of a well defined distance map that takes into consideration the grey-scale information in the image.

Afterwards in Chapter 5, we introduce the eight off-line data sets that will be used during the experimentation. The first data set is an text line extraction competition corpus while the other seven corpora correspond to real historical handwritten text documents. These historical handwritten text corpora present very different layouts covering: normal prose, notarial records, scientific dissemination and music scores. For each of these seven data sets we will review the design configuration considered in order to apply our approach, for the specific region classification required in each of them.

In Chapter 6 we propose specific evaluation measures to assess the accuracy of region detection and classification. We review the properties that correct evaluation measures must have and list current evaluation measures for baseline detection and extraction.

Chapter 7, contains all our experimental evaluation of the proposed framework. By using the scenarios represented by the data sets introduced in Chapter 5 in conjunction with the evaluation

measures listed in Chapter 6 we perform specific experimentation with the aim to prove the hypothesis listed in Chapter 1. More specifically: we formally compare the importance with which the detection and extraction results impact the final HTS result, review various evaluation measures and how they reflect the impact they produce in other systems that use the outputted results they measure, study the added value our detection and classification approach provides, review the importance of language models in layout detection, evaluate the effectiveness of our system in production scenarios, study the effectiveness of combining our approach with Convolutional Neural Networks and finally review the impact of the novel use of automatically generated text content based features.

Finally, Chapter 8 summarizes the contributions of this work, including scientific publications and open sourced software resulting from this work. We finish the thesis with an outlook on the future work that should be performed in the field of HTS.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

PRELIMINARIES

Chapter Outline

1.1 Introduction

Document Layout Analysis (DLA) is a field of pattern recognition that studies the process of identifying and categorizing regions of interest in a scanned page of a document. Research on DLA can be considered to have started the very same day research on Handwritten Text Recognition (HTR) did. The very first Optical Character Recognition (OCR) systems, developed in the late 1960's, that attempted to recognize zip codes on letters or printed information on checks also had to detect and segment the text prior to recognizing it.

Although DLA has evolved into a separate research field in its own right, it is important to note its history and strong relation with HTR related tasks. Furthermore, both research topics are still gaining traction despite their age. This trend can be considered rather shocking, specially if we recognize that printed text recognition (OCR) is regarded a solved task by most researchers in the HTR and DLA communities. This increase in interest is due to two main reasons: current demand to provide access to very large heterogeneous collections of historical handwritten texts and the difficulty in applying the existing methods in such demanding conditions.

This demand to provide access to historical documents, has its origin in the different archives and libraries around the world. By providing this open digitalized access they ensure the preservation of the manuscripts they hold while providing researchers and historical enthusiasts with the means to perform their work.

Unfortunately, more than a mere digitalization of the documents is required in order to provide a truly open and democratic access to the information written in them. The texts needs to be fully transcribed or at least made searchable via an information indexing strategy. This is due to the different calligraphy, grammar, vocabulary, abbreviations, etc... typically used in historical documents that make them inaccessible to most people.

The immense amount of documents held by the aforementioned institutions make the idea of providing these transcriptions via a fully manual transcription unrealistic. In order to resolve this bottle neck, we must either accelerate the transcription process or provide means that allow users to search the document that do not require a full transcription.

It is in this context, that the need for machine learning methods that aid palaeographers or automatize certain aspects of their job has greatly increased. Such learning methods are encompassed in the HTR and DLA research areas.

DLA and more specifically the task of Handwritten Text Segmentation (HTS) is currently very important in order to perform any of the higher level tasks in HTR. Handwritten Text Segmentation is the process by which text regions, at various levels of information (see Fig. 1.2), are detected and extracted. We consider the detection of a region as the identification of its position inside the page. The results of this detection can be provided by means of a simple vertical coordinate, centre of mass, certain momentum values or a baseline in the case of a text line region. By extraction we denominate the process by which the graphical contour around the region is calculated. The difference between the detection and extraction outputs is depicted in Fig 1.1.

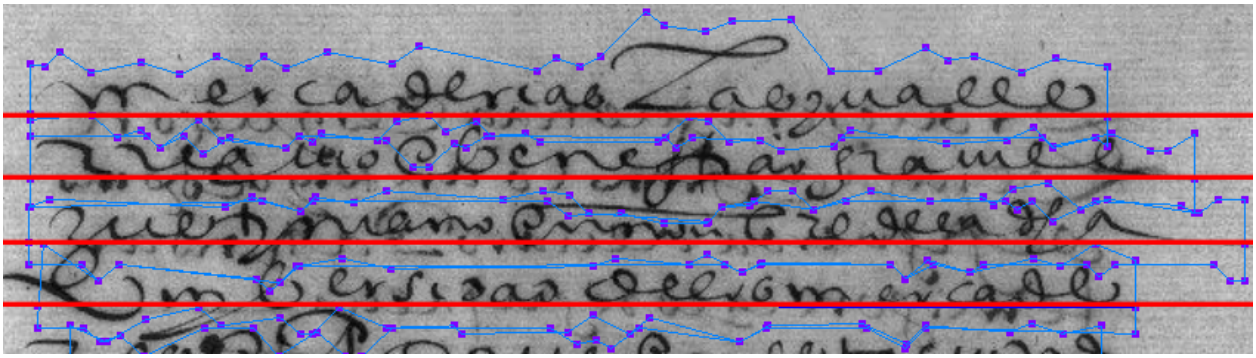


Figure 1.1: Figure representing the results yielded by the detection and extraction phases launched in a text line task. In this case a single vertical coordinate, represented as continuous red line across the page, has been used to annotate the outcome of detection . The extraction result is represented by the blue polygon surrounding each text line.

The importance of HTS is due to the need of currently used automatic [1, 7, 11, 25, 28, 34] or computer assisted [22, 26, 29] transcription systems to be provided with the specific text line images to be transcribed. The segmentation of text lines and even of the words present in those text lines is also required in most key word spotting methods [8, 10, 17, 23, 24] while information indexing methods only require the text lines [2, 18, 31]. Additionaly, text line segmentation is also required in various text alingment techniques [3, 30] used to automatically create additional training corpus from existing non-aligned transcriptions and images of the original documents.

HTS systems have been proven to work adequately on OCR and restricted layout applications. However, they seem lacking in the case of unconstrained handwritten documents; specially when considering historical manuscripts. This is due to the fact that most currently used HTS systems are heuristic based and it has not been until recently that machine learning techniques have been applied, specially in the subtask of text line detection.

We believe that the HTS task is fundamentally a recognition task. At its lowest level of complexity HTS deals with differentiating between the background and foreground classes. classical methods do have an implicit idea of what they are *trying to detect* that can be extracted. Unfortunately, If we inspect the implemented algorithms or the parameters, this implicit idea is not fleshed out and is quite difficult to modify. Additionally, extra knowledge or layout assumptions are hardcoded into them or must be fed into them via manually fine tuned parameters.

Adding to this idea, of the necessity of classes, is the different information levels at which HTS is applied in a page. HTS can be applied in order to target higher level regions like paragraphs, marginalia, diagrams, etc or lover level regions like text lines. We can see examples of both of these types of regions depicted in Fig. 1.2.

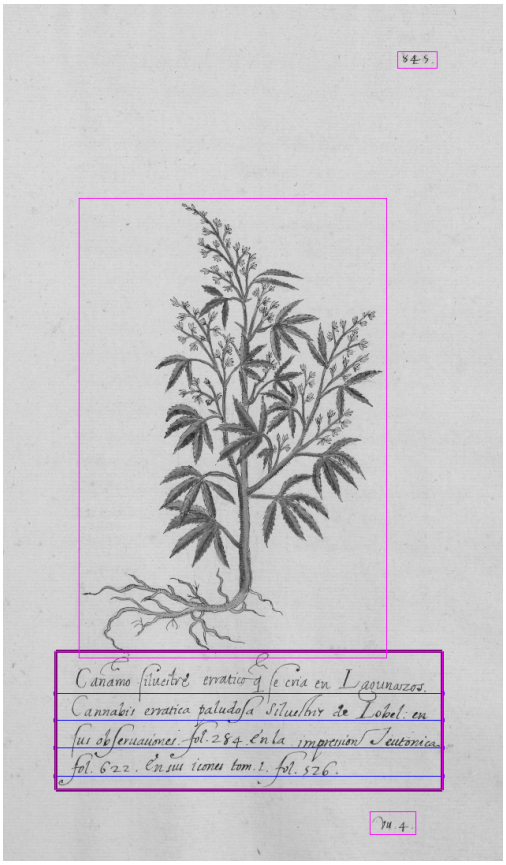


Figure 1.2: This diagram presents the two levels of information at which HTS is applied. The boxed page regions represent the usual targets of HTS applied at a higher information level. The baselines drawn in one of those page regions represent the yielded results of a detection process launched at this lower information level.

Not fleshing out this implicit knowledge is the main origin of the issues that classical methods face when being applied in real production scenarios. The heterogeneity of the layout inside a single collection and the user's needs to target specific regions, that these scenarios present, make it impossible to adapt such techniques in a timely and successful manner.

Additionally, most HTS approaches attempt to both jointly detect and segment the page regions of interest. This tendency is mostly due to the competitions (starting in 2007 [9]) and the promoted evaluation measure adopted by the community for text line segmentation (a subtask of HTS). This *de facto* standard evaluation measures the text line segmentation as per its extraction polygon which incorrectly diminished the importance of the detection subtask. Moreover, the line extraction accuracy results obtained with this measure present little correlation with the transcription accuracy results of the systems using the extracted lines [21].

The main aim of this thesis is to put forward a theoretical HTS framework based on a probabilistic models. This framework will be put into practice by developing and assessing a system that will be production ready and will not present any of the aforementioned issues.

The rest of the chapter is structured as follows; first we perform a small overview of our proposed approach in Section 1.2. In Section 1.3 we perform a small review of the models, formulations and algorithm our approach is based on. Finally, in Section 1.4 we provide a summary of the expected scientific outcomes of this thesis.

1.2 Overview of the Proposed Approach

In order to ensure that we avoid the typical HTS issues covered in the last section, our framework was developed with specific design decisions in mind. First, we consider it crucial to decouple the detection and extraction sub-tasks within HTS. Furthermore, we consider the detection process to be of far more importance than the extraction one. In our opinion the detection task is best tackled when considered a classification task. Thus, we must select an adequate probabilistic framework that allows us to correctly model the intricacies of a page layout.

As per our first design decision we consider that the detection subtask performs the bulk of the work in HTS. Although this opinion is generally accepted by the community it has not been empirically demonstrated to what extent does each sub task contribute to the final accuracy of a segmentation result. In this thesis we will assess the validity of this statement via empirical experimentation.

Our detection subsystem will yield the localization coordinates of the user defined regions found in the specific page. Yielding the detection results as a region baseline, text line baselines in the case of the subtask text line segmentation, has its benefits. It allows us to evaluate independently the accuracy of our detection system and also permits us to easily review and correct results before the extraction phase. This fact will be the cornerstone of the iterative process defined to successfully use the developed technology in real production scenarios. In this thesis we define our semi-automatic iterative process that is able to generate text line segmentation comparable in quality to groundtruth. In this thesis we will present results of how this process fared in various production scenarios as part of the endeavour to transcribe various manuscripts.

This implies that our extraction subsystem will have to use the information provided by the segmentation frontier in order to calculate the extraction polygon around the regions of interest. The separation of the extraction subsystem is already being forced in the Text Line Segmentation (an HTS sub task) in which recently the scientific community has shifted its focus to baseline detection [5, 16]. In this thesis we present our dynamic programming approach based on distance maps. This method is capable of calculating the equidistant extraction polygons of the regions of interest as per the provided region baseline information.

Additionally, this new separation of concerns obliges us to reconsider the adequacy of the current extraction evaluation measures. We must not only evaluate the accuracy of the yielded extraction polygon but also measure and understand how much does our extraction approach enhance the information already provided by the detection system. Another new important aspect to measure is its capability to adequately escalate in performance as the accuracy of the provided

segmentation frontiers improves. In this work, through the use of a basic benchmark method and the groundtruth information, we provide an adequate strategy to measure this.

The biggest conceptual shift, performed by the framework presented in this thesis, is in how it tackles the actual detection of the page regions of interest. HTS systems are usually classified by the order in which they process the different information levels present in a single page [6]. Systems that attack both levels usually have different methods in place for each level. Moreover, some systems are only specially designed to target specific types of page regions: paragraphs, side notes, text lines,... This leaves the user with the task of mixing and matching the different systems in order to segment accurately the regions of their interest. In some cases even forcing them to implement a new approach or version of an existing approach to fit their needs.

Our framework solves this issue by providing users with a direct way to define the classes of interest and the statistical rules that must be followed in order to compose a correct page. The existence of the explicit idea of classes will also provide us with a mechanism to deal with the great variability of different page region types that generally causes poor performance in classical HTS systems.

The method does this by tackling the detection task as a page region classification task. Our approach is based on the idea that a page (or segment of a page) is really a vertical concatenation of regions of several types. By classifying the different regions present in the sequence we obtain as a by-product the localization of each of them.

We train an individual stochastic model for each of the desired vertical region types. Additionally we use a specialized probabilistic model, which we name "vertical layout model" (VLM), to model the concatenation restrictions and prior probabilities of each of the vertical page regions. As we will see later, this model also allows us to easily incorporate human expert knowledge regarding the layout as valuable input for our method.

The technology used to model each page region is based on Hidden Markov Models (HMMs). We use HMMs similarly as how they are used in Handwritten Text Recognition (HTR) and Automatic Speech Recognition (ASR) [13, 19]. The biggest difference between each application being the type of feature vectors sequences we input into the models; In ASR the feature vectors represent acoustic data, line-image features are used as the input sequence for HTR and in our case we use the mass of foreground pixels and its variations as the features to perform HTS. The seminal basic idea of applying HMMs to HTS was introduced in [15].

This machine learning classification based approach will require us to develop new evaluation measures. In this thesis we define the evaluation measures and review their suitability. We take the extra step, usually found missing, to review how they relate to the human understanding of quality and how they correlate to the evaluation measures used in the higher level tasks that depend on the yielded page regions.

Another important topic that will be covered in this thesis is the actual applicability of our framework to real production scenarios. As we have commented previously in this chapter heuristic based methods do not seem adequate to be used for production scenarios. The same has been

said in the past regarding machine learning methods due to the data needs required to achieve reasonable results. Although this concern is currently not seen as a show stopper, it is important we address it. In this thesis we will detail a simple yet effective work-flow to utilize the developed framework in real production scenarios. This process capitalizes on the use of the Expectation Maximization (EM) training algorithm in our approach. The semi-supervised nature of the EM makes our approach have a very simple training data requirement. We will evaluate the applicability of the developed technology in actual production scenarios as part of actual projects.

It is important to accentuate the importance we have given to evaluation measures and their adequacy throughout this thesis. We will start with a study on the actual relevance of extraction evaluation measures. Next, we create of our own classification measures and study how they relate to currently adopted baseline evaluation measures. Finally, we take into consideration performance measures when we apply our framework to real production tasks and relating them to user review time.

In this thesis we also present our novel work on the use *Text Content Features* in HTS. It is usually considered that classification or detection of page regions could greatly improve if some sort of information regarding the contents of each page area was provided. Furthermore, certain types of HTS classification problems can not be performed with only the information provided by graphical features due to the lack of graphical difference between the region types. Unfortunately, having information regarding the contents of the page, classically, has a dependency on detecting the actual areas of interest first. As we will see in this thesis we are able to solve this deadlock by using automatically calculated word probabilistic indexes. The obtained information is used as part of our feature vectors to improve the classification accuracy.

We finalize the outline of the work presented in this thesis by describing our experimentation on the tandem of combination of Convolutional Neural Networks (CNNs) and our developed HMMs approach. Similarly to the work performed in ASR [12] and HTR [4] we apply this tandem combination to our HTS problem. By using the foreground text probability maps yielded by the CNN we are able to eliminate the classical error-prone heuristic preprocess of page images and improve the performance of our system.

1.3 Theoretical Background

This section provides a simple overview of the main theoretical foundations of Hidden Markov Models. As our detection and classification approach is based on these models it is important that we provide certain information regarding them. We will review the theoretical bases, formulations and main algorithms related to HMMs and Language Models.

1.3.1 Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical Markov model where only the emissions and not the states are visible. A HMM can be represented by:

A finite set of states each of which is associated with a probability function that defines or rules the "emissions" in that state.

Transition probabilities which govern the transitions among the states of the HMM.

HMMs are considered one of the most successful models used for Automatic Speech Recognition (ASR) due to the results obtained with them during the past decades. This successes are due to the models ability to represent the problem and its sequential nature in a formal way, hence mathematically tractable, the discrete time sequences of the extracted acoustic feature vectors.

HTR shares one major similarity with ASR as the discrete time sequences that define the continuous writing can be considered also as emissions from an HMM. For HTR the observed emissions represent line-image features and point coordinates of the handwritten pen strokes. Due to this similarity and the success of HMM for the ASR task the application of these statistical models has gained popularity for the resolution of HTR tasks.

For layout detection the same similarity can be considered as the vertical regions that define the page will be emitted by an HMM.

Definition

One possible classification of HMMs can be performed according to the nature of the observed emissions [13, 14]

- If the observed emissions are represented by a vector of symbols in a finite alphabet the HMM can be considered **discrete**.
- When emissions are vectors of reals the HMM is defined as **continuous**.
- **Semi-continuous** is used when the emissions are of a discrete nature but are modelled using continuous probability density functions.

Since in our research we will work only with continuous HMMs we will provide a summarized formal definition of this kind of HMM using the notation represented in [27]. The following assumptions are considered for our continuous HMMs:

- Emissions are only performed in states and not in transitions.

- An additional initial state similar to the end state, which can not perform emissions, has been defined.

Both in ASR and HTR, HMMs are used to compute the probability of the input signal represented as a sequence of feature vectors. Let be $\boldsymbol{x} = \boldsymbol{x}_1\boldsymbol{x}_2...\boldsymbol{x}_T$, a sequence of observations, an HMM model M approximates the probability of this sequence; that is[a]:

$$\Pr(\boldsymbol{x}) \approx P_M(\boldsymbol{x}) \tag{1.1}$$

if the obsertations are reals vectors, we will instead approximate the probability density of $\boldsymbol{x}$.

Formally, a continuous HMM is a finite state machine defined by the sextuple (Q, I, F, X, a, b) where:

- Q is a finite set of states. In order to avoid confusions with the indexation of the different states, we denote the states of the model as $q_0, ..., q_{|Q|-1}$, whereas a sequence of states that generates the vector sequence $\boldsymbol{x} = \boldsymbol{x}_1\boldsymbol{x}_2...\boldsymbol{x}_T$ will be denoted as $z_1z_2...z_T$.
- I is the initial state, an element of Q: $I \in Q$. $I = q_0$
- F is the final state, an element of Q: $F \in Q$. $F = q_{|Q|-1}$
- X is a real d-dimensional space of observations: $X \subseteq \Re^d$.
- a is the state-transition probability function[b]:

$$a(q_i, q_j) = P(z_{t+1} = q_j | z_t = q_i), \quad q_i \in \{Q - \{F\}\}, \quad q_j \in \{Q - \{I\}\}.$$

  Transition probabilities should satisfy $a(q_i, q_j) \geq 0$ and

$$\sum_{q_j \in (Q - \{I\})} a(q_i, q_j) = 1, \quad \forall q_i \in \{Q - \{F\}\}.$$

- b is a probability distribution function[c]:

$$b(q_i, \boldsymbol{x}) = P(\boldsymbol{x}_t = \boldsymbol{x} | z_t = q_i), \quad q_i \in \{Q - \{I, F\}\}, \quad \boldsymbol{x} \in X.$$

  The following stochastic constraints must be satisfied:

$$b(q_i, \boldsymbol{x}) \geq 0,$$

$$\int_{\boldsymbol{x} \in X} b(q_i, \boldsymbol{x}) d\boldsymbol{x} = 1, \quad \forall q_i \in \{Q - \{I, F\}\}.$$

.

[a] "True" probabilities are written as $\Pr(\ldots)$, in contrast with model approximations such as $P_M(\ldots)$ which, to simplify notation, will be denoted as $P(\ldots)$ whenever M can be understood.

[b] $z_t = q_i$ means that the HMM is in the state q_i at time t.

[c] $\boldsymbol{x}_t = \boldsymbol{x}$ means that the HMM in the state z_t generates $\boldsymbol{x}$ at time t.

As the observations are continuous we must therefore use a continuous probability density function. In this case it is defined as a weighted sum of G Gaussian distributions:

$$b(q_j, \boldsymbol{x}) = \sum_{g=1}^{G} c_{jg} b_g(q_j, \boldsymbol{x})$$

where,

$$b_g(q_j, \boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_{jg}|}} e^{(-\frac{1}{2}(\boldsymbol{x}-\mu'_{jg})\Sigma_{jg}^{-1}(\boldsymbol{x}-\mu_{jg}))}$$

- μ_{jg} is the mean vector for the component g of the state q_j
- Σ_{jg} is the covariance matrix for the component g of the state q_j
- c_{jg} is the weighting coefficient for the component g of the state q_j, and should satisfy the stochastic constrain $c_{jg} \geq 0$ and

$$\sum_{g=1}^{G} c_{jg} = 1$$

For the sake of mathematical and computational tractability, the following assumptions are made in the theory of HMMs:

1. **The Markov assumption.** As given in the definition of HMMs, transition probabilities are defined as: $a(q_i, q_j) = P(z_{t+1} = q_j | z_t = q_i)$. In other words it is assumed that the next state is dependent only upon the current state; that is,

   $$\Pr(z_{t+1} | z_1 ... z_t) \approx P(z_{t+1} | z_t)$$

   Which is called the Markov assumption and when applying it to the HMM the resulting model becomes actually a first order HMM.

2. **The stationary assumption.** Assumes that state transition probabilities are independent of the actual time at which the transitions take place. Mathematically,

   $$P(z_{t_1+1} = q_j | z_{t_1} = q_i) = P(z_{t_2+1} = q_j | z_{t_2} = q_i)$$

   for any t_1 and t_2

3. **The output independence assumption.** The probability distribution function is defined as: $b(q_i, \boldsymbol{x}) = p(\boldsymbol{x}_t = \boldsymbol{x} | z_t = q_i)$. This means that the current output (observation) is statistically independent of the previous outputs (observations) and it only depends of the current state; that is,

   $$\Pr(\boldsymbol{x}_t | \boldsymbol{x}_1 ... \boldsymbol{x}_{t-1}, z_1 ... z_t) \approx P(\boldsymbol{x}_t | z_t)$$

Basic algorithms for HMMs

Once we have an HMM, there are three problems of interest. The Evaluation Problem, the Decoding Problem and the Learning Problem.

- The Evaluation Problem. This problem consists in computing the probability $P(\boldsymbol{x})$; that is, the probability that the observations are generated by the model.
- The Decoding Problem. Given a HMM and a sequence of observations $\boldsymbol{x}$, the problem is to find the most likely state sequence in the model which produced the observations. In other words, the problem consists on finding the hidden part of the HMM.
- The Learning Problem. Given a HMM and a sequence of observations $\boldsymbol{x}$, how should we adjust the model parameters in order to maximize the probability $P(\boldsymbol{x})$.

To simplify the notation, in the next sections, $a(q_i, q_j)$ will be written as a_{ij} and $b(q_i, x)$ as $b_i(x)$. Additionally, any kind of subsequence will be represented as $l_i...l_j$ or as $\boldsymbol{l}_i^j$, whenever it is convenient.

The Evaluation Problem and the Forward and Backward Algorithms

Let $\boldsymbol{x}$ be a sequence of real vectors and $Z = \{\boldsymbol{z} = z_1 z_2 ... z_T : z_k = q_i \in (Q - \{I, F\}), 1 \leq i \leq |Q| - 2\}$ a set of state sequences associated with the vector sequence $\boldsymbol{x}$. Then, the probability that $\boldsymbol{x}$ be generated by the HMM is:

$$P(\boldsymbol{x}) = \sum_{\boldsymbol{z} \in Z} \left(\prod_{i=1}^{T} a_{z_{i-1} z_i} b_{z_i}(\boldsymbol{x}_i) \right) a_{z_T F}$$

where z_0 is the initial state I: $z_0 = q_0 = I$.

This calculation involves a number of operations that is in the order of N^T, where N is the number of states of the model excluding the initial state, $N = |Q| - 1$ ($Q = \{q_0 = I, q_1, ..., q_{N-1}, q_N = F\}$), and T is the number of vectors of the sequence. This is very large even if the length of the sequence, T is moderate. Hence, for practical reasons, we must look for another method to perform this calculation.

The **Forward** algorithm is an efficient algorithm which computes $P(\boldsymbol{x})$. The time complexity of this algorithm is $O(|Q|^2 \cdot T)$; however, if we further restrict the HMM topology to using a left-to-right HMM while also limiting the transitions the complexity falls to $O(|Q| \cdot T)$. For the remainder of this thesis we will use the term left-to-right HMM topology to denote a topoly where a transition between two states $q_i, q_j \in Q$ from the HMM, is only possible if $j \geq i$, and where

the number of states it can transition to is reduced so that the resulting transition matrix is a band matrix.

The **Forward** function $\alpha_j(t)$ for $0 < j < N$, is defined as the probability of the partial observation sequence $x_1 x_2 ... x_t$, when it terminates at the state j. Mathematically, $\alpha_j(t) = P(\boldsymbol{x}_1^t, q_j)$ and it can be expressed in the following recursive manner:

$$\alpha_j(t) = \begin{cases} a_{0j} b_j(x_1) & t = 1, \\ \left(\sum_{i=1}^{N-1} \alpha_i(t-1) a_{ij} \right) b_j(\boldsymbol{x}_t) & 1 < t \leq T. \end{cases}$$

with the initial condition that $\alpha_0(1) = 1$. Using this recursion we can calculate the probability that the sequence $\boldsymbol{x}$ be emitted by the model M as:

$$P(\boldsymbol{x}) = P(\boldsymbol{x}_1^T) = \alpha_N(T) = \sum_{i=1}^{N-1} \alpha_i(T) a_{iN}.$$

In a similar way we can define the **Backward** function $\beta_i(t)$ for $0 < i < N$, as the probability of the partial observation sequence $x_{t+1} x_{t+2} ... x_T$, given that the current state is i. Mathematically, $\beta_i(t) = P(\boldsymbol{x}_{t+1}^T | q_i)$ and it can be expressed on a recursive way:

$$\beta_i(t) = \begin{cases} a_{iN} & t = T, \\ \sum_{j=1}^{N-1} a_{ij} b_j(\boldsymbol{x}_{t+1}) \beta_j(t+1) & 1 \leq t < T. \end{cases}$$

with the initial condition that $\beta_N(T) = 1$. Using this recursion the probability that the sequence $\boldsymbol{x}$ be emitted by the model M can be calculated as:

$$P(\boldsymbol{x}) = P(\boldsymbol{x}_1^T) = \beta_0(1) = \sum_{j=1}^{N-1} a_{0j} b_j(x_1) \beta_j(1).$$

As in the forward algorithm the time complexity is: $O(|Q|^2 \cdot T)$, and using a left-to-right HMM the complexity falls to $O(|Q| \cdot T)$.

The Decoding Problem and the Viterbi Algorithm

In this case we want to find the most likely state sequence, $\boldsymbol{z} = z_1 z_2 ... z_T$, for a given sequence of observations, $\boldsymbol{x}$. The algorithm used here is commonly known as the Viterbi algorithm, which maximizes the joint probability of the possible sequence of states given a specific observation sequence; that is $\max_{\boldsymbol{z}} P(\boldsymbol{x}, \boldsymbol{z})$. This algorithm is similar to the forward algorithm, but replacing the sum by the dominating term.

$$v_j(t) = \begin{cases} a_{0j} b_j(x_1) & t = 1, \\ \left(\max_{i \in [1, N-1]} v_i(t-1) a_{ij} \right) b_j(\boldsymbol{x}_t) & 1 < t \leq T. \end{cases}$$

with the condition that $v_0(1) = 1$. Where $v_N(T)$ is $\max_{\boldsymbol{z}} P(\boldsymbol{x}, \boldsymbol{z})$ and using the above defined recursion it can be calculated as:

$$v_N(T) = \max_{i \in [1, N-1]} v_i(T) a_{iN} \leq \sum_{i=1}^{N-1} \alpha_i(T) a_{iN} = \alpha_N(T)$$

The time complexity of the Viterbi algorithm is: $O(|Q|^2 \cdot T)$, and using a left-to-right HMM the complexity falls to $O(|Q| \cdot T)$.

The Learning Problem and the Baum-Welch Algorithm

The learning problem is how to adjust the HMM parameters ($a_{ij}, b_i(x), c_{jg}, \mu_{jg}$ and Σ_{jg}), so that a given set of observations (called training set) is generated by the model with maximum likelihood. The Baum-Welch algorithm [20] (also known as Forward-Backward algorithm), is used to find these unknown parameters. It is an expectation-maximization (EM) algorithm.

Let $E = \{\boldsymbol{x}_r = \boldsymbol{x}_{r1}\boldsymbol{x}_{r2}...\boldsymbol{x}_{rT_r} : \boldsymbol{x}_{rk} \in X, 1 \leq k \leq T_r \wedge 1 \leq r \leq R\}$ a set of R vector sequences, used to adjust the HMM parameters. The basic formula to estimate the state-transition probability a_{ij} is:

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r - 1} \alpha_i^r(t) a_{ij} b_j(\boldsymbol{x}_{rt+1}) \beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)}$$

where $0 < i < N$, $0 < j < N$ and $P_r = P(\boldsymbol{x}_r)$ is the total probability of the sample r from set E.

If the probability density function of each state on the HMM is approximated by a weighted sum of G Gaussian distributions we must find the unknown parameters c_{jg}, μ_{jg} and Σ_{jg}. With this purpose we define $L_{jg}^r(t)$ as the probability that the vector $\boldsymbol{x}_{rt} \in \Re^d$ be generated by the Gaussian component g in the q_j state:

$$L_{jg}^r(t) = \frac{1}{P_r} U_j^r(t) c_{jg} b_{jg}(\boldsymbol{x}_{rt}) \beta_j^r(t)$$

where

$$U_j^r(t) = \begin{cases} a_{0j} & if \quad t = 1, \\ \sum_{i=1}^{N-1} \alpha_i^r(t-1) a_{ij} & otherwise. \end{cases}$$

Taking into account the previous definitions, the parameters c_{jg}, μ_{jg} and Σ_{jg} can be estimated as:

$$\hat{\mu}_{jg} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jg}^r(t) \boldsymbol{x}_{rt}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jg}^r(t)}$$

$$\hat{\Sigma}_{jg} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jg}^r(t) (\boldsymbol{x}_{rt} - \hat{\mu}_{jg})(\boldsymbol{x}_{rt} - \hat{\mu}_{jg})'}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_{jg}^r(t)}$$

$$c_{jg} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jg}^{r}(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{j}^{r}(t)}$$

The time complexity of one iteration of the Baum-Welch algorithm is: $O(R\cdot|Q|^2\cdot T)$; however, using a left-to-right HMM the complexity falls to $O(R\cdot|Q|\cdot T)$. This algorithm is iterated until some convergence criterion is reached.

Sometimes, it is necessary to have a composition of C HMM joined sequentially, for example in the case of the different vertical regions that conform a page. In this case, the "embedded training Baum-Welch" algorithm, which re-estimates the parameters of the composition of C sequentially concatenated HMMs, can be used. This algorithm enables to train the HMM without any prior segmentation of the training page images into vertical regions. In [28] we can find all the formula to compute the unknown parameters in this case but more specifically for the HTR case (lines and morphemes).

1.3.2 Language Models

Language models (LMs) are usually used to model text properties, like syntax and semantic, independently from the character morphology modelled by HMMs. They are used in many natural language processing applications such as speech recognition, machine translation or handwritten text recognition. These models can be used to predict the next word in a word sequence.

In our research LMs are used in order to model the structure of a page as described by the composition of the different regions / text sections that can compose it. In the document layout analysis field the words classically predicted by the LM are the regions, and the full sentences can be equated to an adequate region composition that makes a page. Given a sequence of regions $\boldsymbol{x} = x_1, x_2, ..., x_m$, we can use the chain-rule to exactly decompose the probability of this sequence $\boldsymbol{x}$ as:

$$\Pr(\boldsymbol{x}) = \Pr(x_1)\cdot\prod_{l=2}^{m}\Pr(x_l|\boldsymbol{x}_1^{l-1})$$

where $\Pr(x_l|\boldsymbol{x}_1^{l-1})$ is the probability of the region x_l when we have already seen the sequence of regions $x_1...x_{l-1}$. The sequence of regions prior to x_l is called history.

In practice, for HTR, estimating the probability of sequences can become difficult since sentences can be arbitrarily long and hence many sequences are not observed during LM training. It is necessary to note that for a vocabulary with $|V|$ different words, the number of different histories is $|V|^{l-1}$. So, the estimation of $\Pr(\boldsymbol{x})$ can be unworkable. For that reason these models are often approximated using smoothed n-gram models, which obtain surprisingly good performance although they only capture short term dependencies.

This LM estimation issues do not usually occur in the case of layout detection, considering the allowed region types. The low number of types ensures that samples of transitions from one region to the rest can be usually found. Regardless of this, estimation of the LM in layout detection still benefits from the use of smoothed n-gram models.

The n-gram approximation makes the assumption that the probability of a symbol depends only on the $n-1$ previous symbols; that is:

$$Pr(\boldsymbol{x}) \approx \prod_{l=1}^{m} \Pr(x_l|\boldsymbol{x}_{l-n+1}^{l-1}) \tag{1.2}$$

Owing to the fact that $l-n \leq 0$ for the first $n-1$ words in $\boldsymbol{x}$, Eq. (1.2) must be written as:

$$\Pr(\boldsymbol{x}) \approx P(x_1) \cdot \prod_{l=2}^{n-1} P(x_l|\boldsymbol{x}_1^{l-1}) \cdot \prod_{l=n}^{m} P(x_i|\boldsymbol{x}_{l-n+1}^{l-1})) \tag{1.3}$$

Another benefit of n-grams is that they can be easily learnt from training data. Given a vocabulary of region labels Σ and the region labels training data or layout corpora represented by $\boldsymbol{w} = w_1 w_2 ... w_l$, the estimated probability of the region $a \in \Sigma$, having seen a sequence of $n-1$ regions $\boldsymbol{z} \in \Sigma^{n-1}$, is computed as:

$$P(a|\boldsymbol{z}) = \frac{f(\boldsymbol{z}a)}{f(\boldsymbol{z})}$$

where $f(\boldsymbol{z})$ is the number of times that the sequence $\boldsymbol{z}$ has appeared in the training sequence $\boldsymbol{w}$. This is a maximum likelihood (ML) estimate.

n-grams Modelled by a Stochastic Finite State Automaton

Along this work, stochastic finite state automata (SFSA) are often used to represent HMMs, lexical models and language models. Thanks to the homogeneous finite-state nature of all these models, they can be easily integrated into a single global finite state model.

A SFSA is usually defined as a sextuple $A = (Q, \Sigma, \delta, q_0, P, F)$, where:

- Σ is non-empty finite set of symbols
- $Q \subseteq \Sigma^{n-1} \cup q_0$ is a finite, not-empty set of states.
- $\delta \subset Q \times \Sigma \times Q$ is the state-transition function.
- q_0 is the initial state ($q_0 \in Q$)

- $P : \delta \rightarrow \Re^+$ is the probability transition function. We are using deterministic SFSA, so each transition is identified with only the source state $q \in \Sigma^{n-1}$ and the transition symbol $v \in \Sigma$. Therefore, $P(q, v, q') = P(v|q)$
- $F : Q \rightarrow \Re^+$ is the final state probability function.

Given an specific n-gram model trained on $\boldsymbol{w} = w_1 w_2 ... w_l$ region label corpus we can adapt the SFSA [32, 33] definition to represent it as follows:

- Each state in Q is defined using the region vocabulary symbols Σ with a definition length of up to n symbols. A state $q = (w_{l-n+1} \dots w_{l-2} w_{l-1})$[d] will exist to represent all the sequences of length n in $\boldsymbol{w}$ having states with up to $n-1$ symbols for the initial $n-1$ regions of w.
- The state transition function δ represents a shift in the n sized window on the corpus $\boldsymbol{w}$. Given the sequence $w_{l-n+1} w_{l-n+2} \dots w_{l-2} w_{l-1} w_l \in w$ there will exist two states in Q representing the previous context $q_{l-1} = (w_{l-n+1} w_{l-n+2} \dots w_{l-2} w_{l-1})$ and the current context $q_l = (w_{l-n+2} \dots w_{l-2} w_{l-1} w_l)$, and a transition to represent the shift from one to the other:

$$(w_{l-n+1} w_{l-n+2} \dots w_{l-2} w_{l-1}, w_l, w_{l-n+2} \dots w_{l-2} w_{l-1} w_l)$$

- The initial state q_0 represents the empty string λ

In Fig. 1.3 we can see sample SFSAs that represent different generic $n - gram$ models. Additionally, in Fig. 1.4 we provide sample automata that have been trained with the same sample corpus to represent different n-gram models.

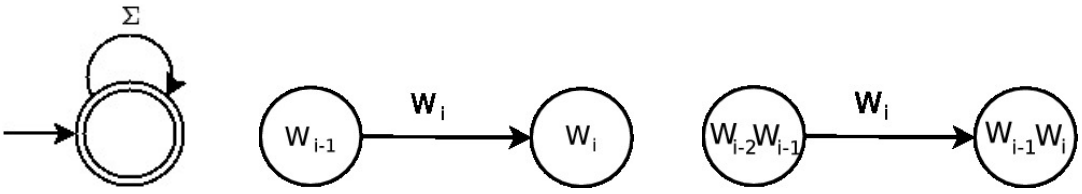


Figure 1.3: Examples of n-grams represented using a SFSA.

[d] Which would represent the history prior to reaching symbol w_l

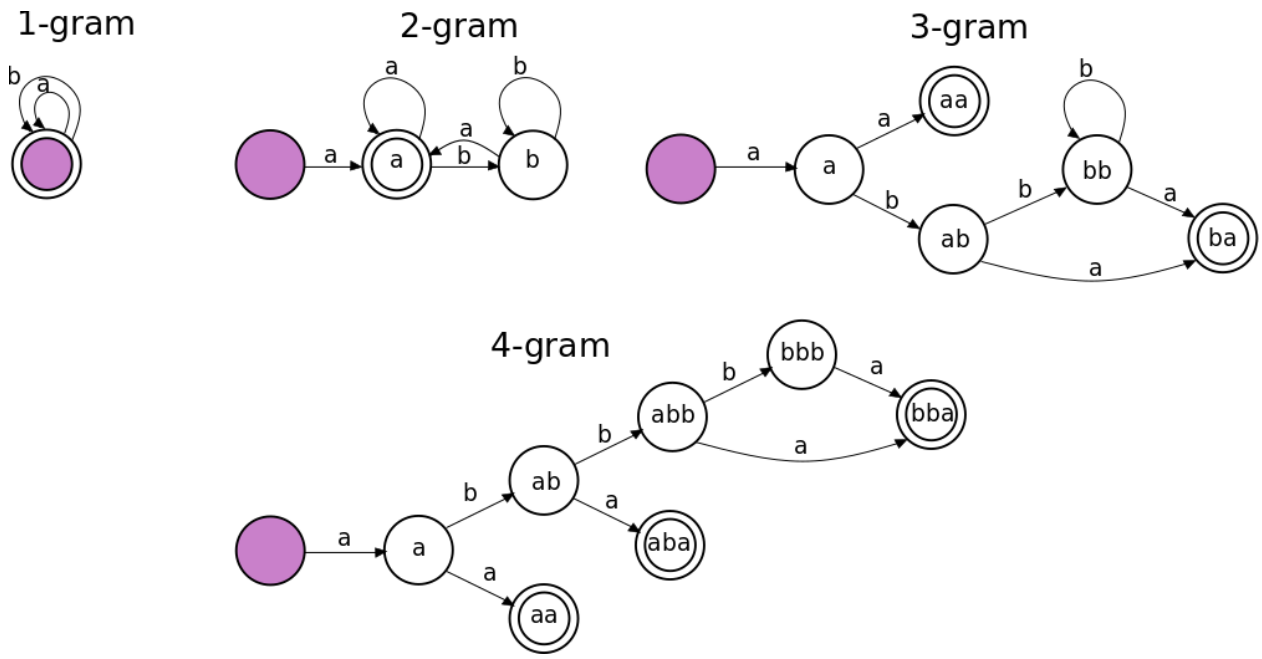


Figure 1.4: Examples of different SFSA trained over the same corpus $\boldsymbol{w} = \{aa, aba, abba, abba\}$ equivalent to the different n-gram models for $n = [1, 4]$

1.4 Expected Scientific Outcomes

The main purpose of this thesis is to present our *Handwritten Text Segmentation* theoretical framework, develop it, and assess it in different scenarios. More specifically, the expected scientific outcomes of this thesis is summarized in the following points:

1. Put forward a theoretical framework to apply machine learning to the *Handwritten Text Segmentation* task and develop a usable system. The intention of this study is to test the applicability of Hidden Markov Model based technology, previously applied to Automatic Speech Recognition and Handwritten Text Recognition, to the Handwritten Text Segmentation problem.

2. The importance of decoupling the detection and extraction sub-phases of HTS. One of the major issues with current HTS systems is produced by the strong coupling of its two subtasks, and the fact that not enough importance was given to the detection phase. One of the objectives of this thesis is to study the impact each of the phases has on the final segmentation accuracy results.

3. Being able to select the type of regions to segment is crucial for a production ready system. Unfortunately, classical methods do not give users the functionality to perform such a selection in an easy manner. In this work we study the importance of tackling the page region detection problem as a classification task that yields the location coordinates as a sub-product.

4. We provide an extraction sub-task method that is able to suitably use the information provided by the detection system. The developed region extraction algorithm uses a specific distance map in order to generate equidistant extraction frontiers.

5. The aforementioned change of emphasis to classification and detection of page regions requires that we develop adequate measures to assess the accuracy of our methods. In this thesis we present our evaluation measures and study how they relate to the accuracy measures of higher level systems that are dependant on our yielded results and also to user effort. Furthermore, we compare our evaluation measures to other currently used measures in order to review their suitability.

6. We will study how the overall system accuracy will improve through the inclusion of a higher order vertical layout model. This model indicates how the different types can be combined to form a correct page. Additionally, this model will allow us to easily add expert knowledge to the detection process.

7. A fully developed system should provide an adequate work-flow in order to tackle real production scenarios. One of the objectives of this work is to showcase a semi-automatic process designed to apply the developed technology to generate groundtruth quality region segmentation when tackling unseen corpora. We study the initial entry cost and scalability of this process.

8. We review the possibility of adding Text Content Features in order to improve accuracy of classification results. Classically, the use of page content information in order to improve accuracy in document analysis tasks was impossible. With the current development of systems that can automatically calculate the word probability for a given page we can by-pass this restriction. In this work we submit a novel strategy to use the information provided by the word probabilistic indexes to create text content features. These text content features improves the classification accuracy in corpora in which the page regions of interest do not present a clear graphical difference.

9. Study of the tandem combination of CNNs and HMMs will be performed. The CNNs will be used in order to yield a probability map of the text foreground which will be used as features for our HMMs. This tandem combination will prove to be beneficial as it eliminates the need of classical historical text image preprocessing techniques.

10. Lastly, the resulting systems for the defined classification and extraction approach will be shared freely. In this way the designed theoretical frameworks in this work can be fully tested in practice and the results replicated by other researchers.

As we will see in Chapter 8, all of the expected outcomes have been adequately fulfilled. We developed a a page region classification and detection system based on HMMs and finite state automata vertical layout models. This system was successfully tested in a large number of corpora in research and production scenarios for both text line and page region areas. The classification system has been proven to considerably reduce the effort required by an user in order to produce groundtruth quality region detection. We have successfully developed a region extraction frontier

algorithm that adequately uses the information yielded by the region detection system. The extraction system is able to calculate equidistant frontiers and its accuracy is proportional to the accuracy of the provided region or textline baselines. We have created our own evaluation measures and have performed experiments that relate them to user effort and the accuracy of the higher level systems that are dependant on the results of HTS systems. Furthermore, we have reviewed in depth the current status of evaluation measures used in handwritten text segmentation. We have introduced the novel idea of text content features via the use of word probabilistic indexes and showed the beneficial impact they have. Finally, we have applied CNNs in tandem with our existing HMM based framework to study how they impact the overall accuracy of our system.

1.5 Chapter Conclusions

In this chapter I have introduced the Document Layout Analysis field and how it relates to other pattern recognition fields. I reviewed the current importance of the Handwritten Text Segmentation task the different information levels at which it can be applied and the different types of solutions that can be yielded by the methods.

I provided my thoughts on the design issues that are inherent to classical HTS methods and performed an overview of our own approach that attempts to avoid them. The theoretical background, which is the foundation for our approach, has been provided. Finally, I listed the expected scientific outcomes of this thesis that will be fulfilled during its course.

Bibliography

[1] Bazzi, I., Schwartz, R., and Makhoul, J. (1999). An omnifont open-vocabulary OCR system for English and Arabic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):495–504.

[2] Bluche, T., Hamel, S., Kermorvant, C., Puigcerver, J., Stutzmann, D., Toselli, A. H., and Vidal, E. (2017). Preparatory kws experiments for large-scale indexing of a vast medieval manuscript collection in the himanis project. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 311–316.

[3] Bluche, T., Moysset, B., and Kermorvant, C. (2014). Automatic line segmentation and ground-truth alignment of handwritten documents. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 667–672.

[4] Bluche, T., Ney, H., and Kermorvant, C. (2013). Tandem hmm with convolutional neural network for handwritten word recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2390–2394.

[5] Diem, M., Kleber, F., Fiel, S., Gruning, T., and Gatos, B. (2018). cbad: Icdar2017 competition on baseline detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1355–1360.

[6] Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2017). A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, 64:1 – 14.

[7] Espana-Boquera, S., Castro-Bleda, M. J., Gorbe-Moya, J., and Zamora-Martinez, F. (2011). Improving offline handwritten text recognition with hybrid hmm/ann models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):767–779.

[8] Frinken, V., Fischer, A., Manmatha, R., and Bunke, H. (2012). A Novel Word Spotting Method Based on Recurrent Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):211 –224.

[9] Gatos, B., Antonacopoulos, A., and Stamatopoulos, N. (2007). Handwriting segmentation contest. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1284–1288.

[10] Ghorbel, A., Ogier, J.-M., and Vincent, N. (2015). A segmentation free word spotting for handwritten documents. In *Proc. of the 13th Intl. Conf. on Document Analysis and Recognition (ICDAR'15)*, pages 346–350.

[11] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868.

[12] Hermansky, H., Ellis, D. P. W., and Sharma, S. (2000). Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 3, pages 1635–1638 vol.3.

[13] Jelinek, F. (1998). *Statistical methods for speech recognition*. MIT Press.

[14] Lee, K. (1989). *Automatic speech recognition: the development of the SPHINX system*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers.

[15] Lu, Z., Schwartz, R., and Raphael, C. (2000). Script-independent, hmm-based text line finding for ocr. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 551 –554 vol.4.

[16] Murdock, M., Reid, S., Hamilton, B., and Reese, J. (2015). Icdar 2015 competition on text line detection in historical documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1171–1175.

[17] Pratikakis, I., Zagoris, K., Gatos, B., Puigcerver, J., Toselli, A. H., and Vidal, E. (2016). Icfhr2016 handwritten keyword spotting competition (h-kws 2016). In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 613–618.

[18] Puigcerver, J., Toselli, A., and Vidal, E. (2015). A new smoothing method for lexicon-based handwritten text keyword spotting. In Paredes, R., Cardoso, J. S., and Pardo, X. M., editors, *Pattern Recognition and Image Analysis*, volume 9117 of *Lecture Notes in Computer Science*, pages 23–30. Springer Int. Publishing.

[19] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[20] Rabiner, L. R. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

[21] Romero, V., Sánchez, J. A., Bosch, V., Depuydt, K., and de Does, J. (2015). Influence of text line segmentation in handwritten text recognition. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 536–540.

[22] Romero, V., Toselli, A., and Vidal, E. (2012). *Multimodal Interactive Handwritten Text Transcription*. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing.

[23] Roy, P. P., Rayar, F., and Ramel, J.-Y. (2015). Word spotting in historical documents using primitive codebook and dynamic programming. *Image and Vision Computing*, 44:15 – 28.

[24] Rusinol, M., Aldavert, D., Toledo, R., and Lladós, J. (2015). Efficient segmentation-free keyword spotting in historical document collections. *Pattern Recognition*, 48(2):545–555.

[25] Sánchez, J. A., Romero, V., Toselli, A. H., and Vidal, E. (2016). Icfhr2016 competition on handwritten text recognition on the read dataset. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 630–635.

[26] Toselli, A., Vidal, E., and Casacuberta, F. (2011a). *Multimodal Interactive Pattern Recognition and Applications*. Springer, 1st edition edition.

[27] Toselli, A. H. (2004). *Reconocimiento de Texto Manuscrito Continuo*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia (Spain). Advisor(s): Dr. E. Vidal and Dr. A. Juan (in Spanish).

[28] Toselli, A. H., Juan, A., Keysers, D., González, J., Salvador, I., H. Ney, Vidal, E., and Casacuberta, F. (2004). Integrated Handwriting Recognition and Interpretation using Finite-State Models. *IJPRAI*, 18(4):519–539.

[29] Toselli, A. H., Romero, V., Pastor, M., and Vidal, E. (2009). Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1824–1825.

[30] Toselli, A. H., Romero, V., and Vidal, E. (2011b). *Language Technology for Cultural Heritage*, chapter Alignment between Text Images and their Transcripts for Handwritten Documents., pages 23–37. Theory and Applications of Natural Language Processing. Springer,. Caroline Sporleder, Antal van den Bosch y Kalliopi Zervanou (Eds.).

[31] Toselli, A. H., Vidal, E., Romero, V., and Frinken, V. (2016). HMM Word graph based keyword spotting in handwritten document images . *Information Sciences*, 370-371:497 – 518.

[32] Vidal, E., Thollard, F., C. de la Higuera, F. C., and Carrasco, R. (2005a). Probabilistic finite-state machines - part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025.

[33] Vidal, E., Thollard, F., C. de la Higuera, F. C., and Carrasco, R. (2005b). Probabilistic finite-state machines - part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1025–1039.

[34] Vinciarelli, A., Bengio, S., and Bunke, H. (2004). Off-line recognition of unconstrained handwritten texts using hmms and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720.

CHAPTER 2

RELATED WORK

Chapter Outline

The Document Layout Analysis (DLA) research field and specially its Handwritten Text Segmentation (HTS) sub-field are bountiful in regard to the number of solutions and studies being published. Since the first OCR system developed in the 1960's, hundreds of solutions for document layout analysis have been developed. Additionally, we can observe this progress through the publications found in journals, conferences and books published on the matter. Since the publication of the Run-Length Smoothing Algorithm (RLSA) [29] in 1982 , which can be considered one of the initial HTS publications, hundreds of works have been produced on this topic.

Throughout the years several surveys have been performed [8, 13, 14, 19, 20, 24] to compile this work and classify the underlying strategies used to tackle this task. I do not intend to provide a survey as complex and detailed as the aforementioned articles. However, will provide a review regarding the current status of the field in relation with the work carried out in this thesis.

The rest of this chapter is structured as follows. First, I will provide a list of the considered different information sources. Second, different taxonomies will be introduced, as per our vision, in which to organize the different methods. Finally I will review the state of the art in the specified categories.

2.1 Journals, Conferences and Competitions

In this section I will list the journals, conferences and competitions that are of interest to the DLA community and impact, with their publications, the status of the HTS field. Although I will not be able to review exhaustively each of them it is is important that they are listed in order to set the context of the investigation presented in this thesis.

The are four journals of special interest for the HTS task: International Journal on Document Analysis and Recognition (IJDAR), Pattern Recognition (PR), Pattern Recognition Letters (PRL) and IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).

The following list enumerates the major DLA conferences: International Conference on Document Analysis and Recognition (ICDAR), International Conference on Frontiers in Handwriting Recognition (ICFHR), International Workshop on Document Analysis Systems (DAS), ACM Symposium on Document Engineering (DocEng), International Conference on Computer Vision (ICCV), International Conference on Pattern Recognition (ICPR) and the European Conference on Computer Vision (ECCV).

It is also very important to list the handwritten text segmentation competition. Via the evaluation measures enacted and the data sets they have shared, these competitions have shaped the the HTS community:

- **HTS** Handwriting Segmentation Contest. Long time running competition (2007-2013) that focused on graphical text line extraction accuracy as an evaluation measure. The competition was performed over a synthetic multi-script multi-writer corpus.

- **ANDAR-TL** ANncestry Document Analysis and Recognition Text Lines. A shortly lived competition (2015) that focused on the detection of the origin point of each text line present in a page. The competition corpus was composed by a set of pages taken real records of different sources from the he eighteenth and nineteenth century. Although this competition did not continue it marked a change in how evaluation of text segmentation was performed as it gave more importance to the detection of the line starts.
- **CBAD** Competition on BAseline Detection. The youngest competition (2017-2019) that has taken over the mantle of the HTS competition. This competition showcases the continued focus shift of the community from line segmentation to line detection as a more accurate way to measure HTS. This event has two difficulty tracks with pages selected from actual historical corpus.

2.2 Taxonomies of Handwritten Text Segmentation Problems and Approaches

In order to classify properly the different methods for HTS we must initially define both the *Problem* typology and the *approaches* taxonomy. Both classifications, which are orthogonal, are necessary in order to adequately indicate the traits of the developed techniques.

2.2.1 Taxonomy of Problems

HTS is a task that is usually applied at two different levels on the same document: Page Regions and Text Lines. These levels of application are currently only applied considering single pages of a document. At the larger page region level of application, HTS tackles the detection and extraction of the larger structures of a page such as: paragraphs, side notes, diagrams, text columns, etc. The lower level of HTS considers smaller entities such as lines or words[a]. We can observe the levels and the relation they have with the rest of the layout structure of a collection in Fig. 2.1.

These two levels are also implicitly referenced/used in the classical typologies used to organize the different methods of the aforementioned survey articles: top-down, bottom-up and hybrid. Furthermore, this basic categorization mainly refers to "the order in which the information is processed" [8] or to be more exact the order each approach assumes in order to work properly. In fact, most techniques actually only yield answers for either the *top* or *bottom* level. In this way we can classify the HTS approaches depending on which of the levels of the task they solve:

[a]Personally, I consider word segmentation to be a problem that is actually implicitly solved in many state-of-the-art holistic handwritten text recognitions approaches and therefore should not be considered as a task for HTS

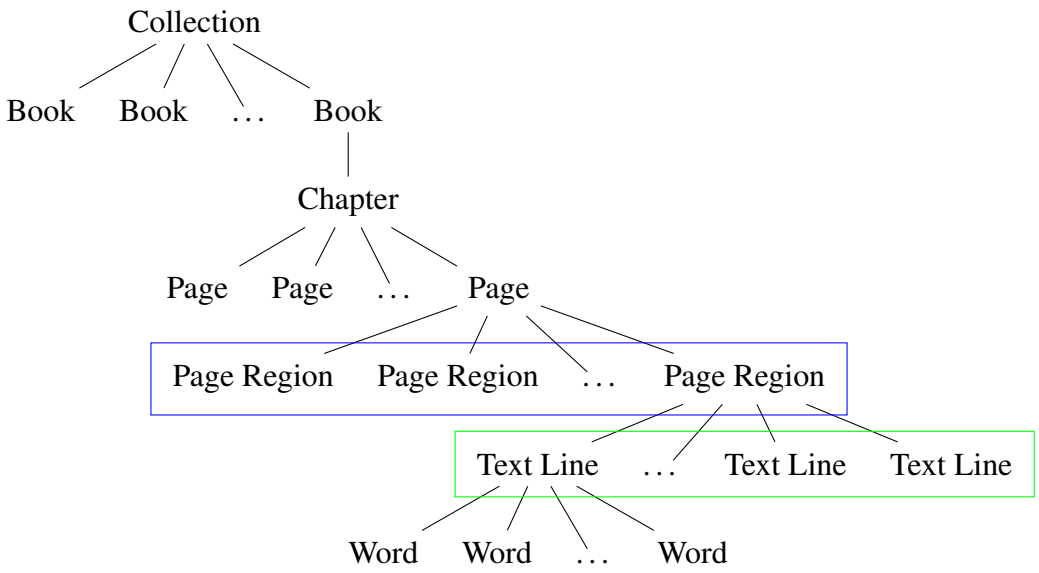


Figure 2.1: Tree structure and composition order representation of a collection. Within this structure we observe the two main levels at which HTS is currently applied: the page region level and the text line level (boxed respectively in different colours inside the diagram).

- **Top or Text Region** - only performs the segmentation of higher order regions present in the text page.
- **Bottom or Text Line** - only performs segmentation of the text lines.
- **Both** - performs both region and text line segmentation. This class can also be sub-divided into:
  - **Sequential** - can perform the segmentation of both levels but only one after the other or can only be applied for a single level of HTS at a time.
  - **Concurrent** - performs the segmentation of both levels at the same time.

2.2.2 Taxonomy of Approaches

Contrary to what is specified in some surveys [8] we believe that the main trait by which to classify HTS approaches is with regard of their use of *Machine Learning* methods. Thus we divide the taxonomy into two main categories: *Heuristic Based* and *Probabilistic*.

Heuristic Based

In our opinion, *Heuristic based* approaches characterize themselves by having a hard-coded underlying set of assumptions and rules that dictate the type of layouts they can be applied to. Although

certain methods do allow, via parameter tuning, a broader set of layouts to be detected; the variation is rather finite and still anchored around an expected layout structure.

Additionally, these approaches do not have generally the concept of classes at the root of their proposed method. The types of regions these techniques detect and segment are codified into the algorithm. This makes it pretty hard or impossible to adapt to user requirements.

Within the *Heuristic based* approaches we would consider the subtypes shown in Fig. 2.2, using the categories described in the latest survey by Eskenazi et al [8]

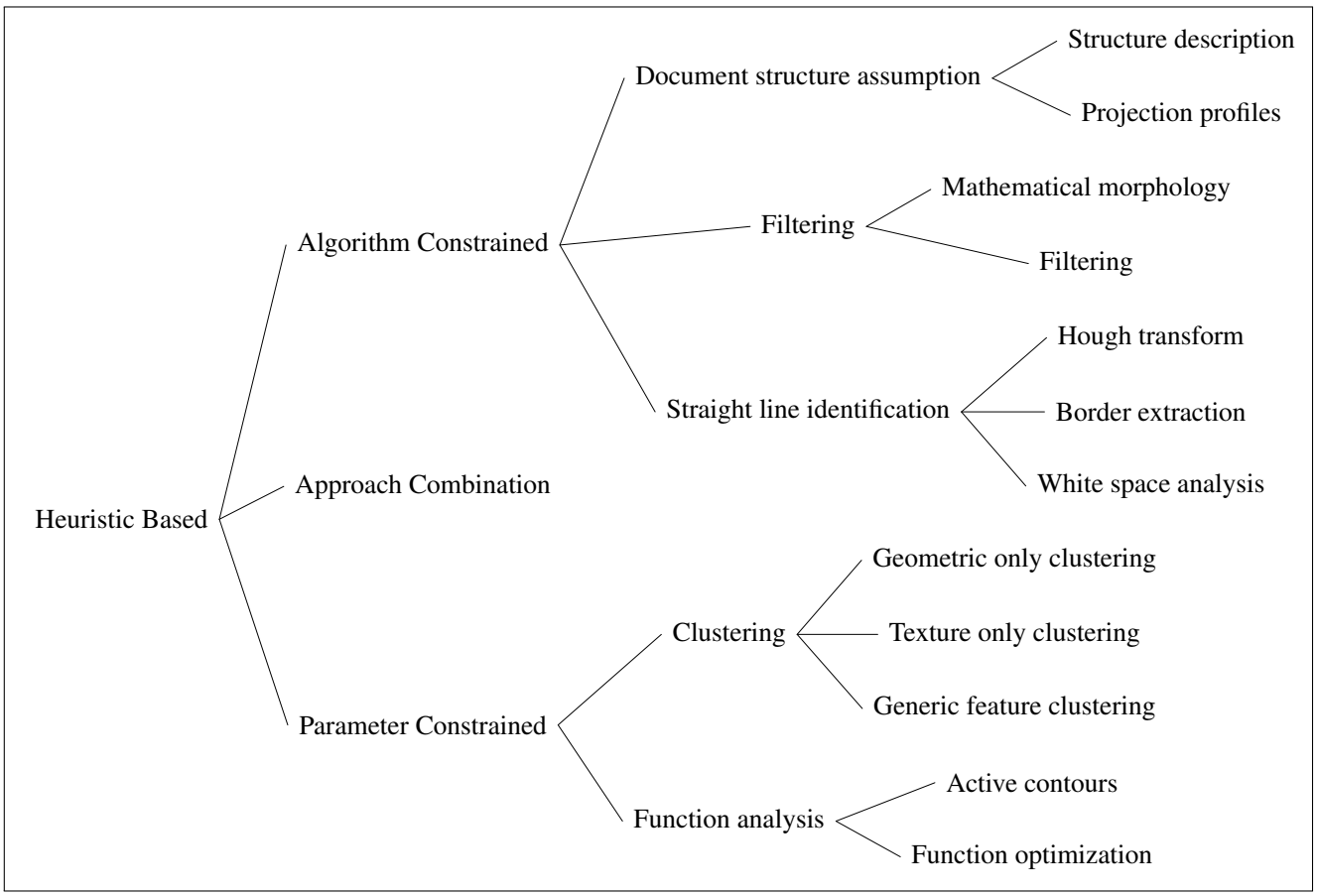


Figure 2.2: Taxonomy of Heuristic Based HTS algorithms.

It is important to note that although Fig. 2.2 resembles other published taxonomies [8] I have performed some changes:

1. I have removed the Probabilistic layout estimation category and the whole classification subtree as I consider them part of the *Probabilistic* methods. We find the differentiation between these categories and the Neural Network methods a glaring issue.

2. I have added the *Approaches Combination* category as part of the *Heuristic Based* techniques as I consider that any combination of other heuristic approaches will still remain in the same category.

Probabilistic

Probabilistic approaches characterize themselves for using Machine Learning models. These methods learn something from training examples of documents and/or page regions they want to detect and segment. Although these types of approaches can be easily adapted to user requirements or new layouts, by training with the adequate data, they are ultimately constrained by the assumptions made by stochastic framework they have chosen.

I do not share the opinion of other surveys where they make a differentiation between terms that are quite difficult to disambiguate: *Probabilistic layout estimation*, *Classification based Segmentation* and *Neural Networks*. If one reads the articles presented as examples of such groups it seems rather difficult to explain why an article would fall into one category or the other. Specially as a neural network can be trained to be a classifier and both concepts can also be seen as yielding estimates of the posterior probability of the layout elements having seen the image.

It is also important to note that some (if not most) of these machine learning based methods have a need to certain heuristic post processing techniques. This is due to the fact that the result yielded directly by the machine learning framework, does not conform to the expected input by the user or system that depends on the HTS results. For example, certain methods have as an output format a posteriori map of the page areas that contain text line pixels while HTR systems usually require an actually extracted text line image. This forces these type of approaches to develop an heuristic method to decide the text line contours or baselines given the yielded probability maps. Although this post processing techniques try to keep assumptions at a minimum, approaches that do not require this step are being researched.

As shown in Fig. 2.3 I mainly divide the *Probabilistic* category into two sub-types. This binary division is performed taking into consideration if the method has the possibility to *easily include Prior* information.

2.3 State of the Art

After defining the different taxonomies that can be found in the HTS approaches and problems taxonomies I now proceed to list the major proposals that I have explored during the course of this thesis. It is important to note throughout the duration of this thesis two major changes have taken place in the HTS community:

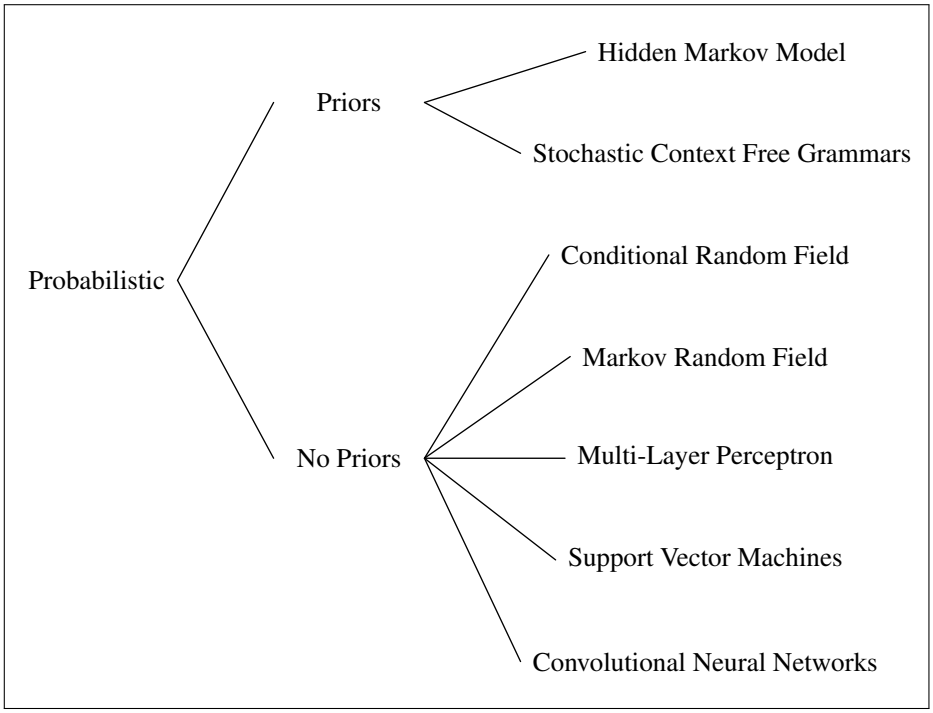


Figure 2.3: Taxonomy of Probabilistic Based HTS algorithms.

- A shift in Text Line Detection methods from contour extraction to baseline detection as the output. This has also implied a major shift in the way accuracy is measured. We will explore this later.
- An evolution towards *Probabilistic* methods.

Based on clustering there is a relatively new method that focuses on the classification and clustering of baseline points [9]. Another effective method uses the detection and clustering of *super points* to detect baselines in pages [11].

Support Vector Machines (SVMs) were more typically used to classify the page pixels into different page regions. Foreground classification [5], distinction between print and handwritten text [7], or divide between different photography, printed text and handwritten text [27]. Wei et al [28] also performed a comparison between SVMs, Gaussian Mixture Models (GMMs) and Multi Layer Perceptrons (MLPs) with no clear winner.

Usage of Conditional Random Fields (CRF) was tried out in different articles but it did not fare well in comparison to Stochastic Context Free Grammars [2, 6]. Markov Random Fields have seen minimal use to differentiate between printed text and handwritten text [21].

The use of Convolutional Neural Networks (CNNs) for HTS tasks has seen a great increase. These articles can be sub-classified as those which only output a probability map [3, 4, 16, 30] and those that use this map to actually generate specific region and baseline results that are more

readily usable by other systems [11, 22, 23]. CNNs can be trained to target both the problem at the larger region or at the text line detection level. Additionally, certain approaches are know to adequately attack both levels in a concurrent manner [22].

Our approach, on region detection and classification, presented in this thesis, can be applied to both the region and line levels in a sequential manner. Our approach is based on Hidden Markov Models extending an early article by Zhidong et al [15].

The method developed to calculate the extraction contour requires the vertical coordinates of the target regions to extract to be provided. Since I have decoupled the detection and extraction sub-phases it does not seem adequate to compare this method to complete HTS approaches. Having said this, the way we calculate the extraction frontier is akin to other techniques that solve this issue based on A^* path finding algorithms [1, 10, 12, 17, 18, 25, 26].

2.4 Chapter Conclusions

In this chapter I have reviewed the current state of the art of document layout analysis. I provided my own reviewed taxonomy of the methods from the solution and problem point of view. Finally I listed the major solutions considered of each method and provided a definition of our approach as per the taxonomies defined in this chapter.

Bibliography

[1] Adiguzel, H., Sahin, E., and Duygulu, P. (2012). A hybrid for line segmentation in handwritten documents. In *Proocedings of ICFHR*, pages 503–508.

[2] Álvaro, F., Cruz, F., Sánchez, J.-A., Terrades, O. R., and Benedí, J.-M. (2013). Page segmentation of structured documents using 2d stochastic context-free grammars. In Sanches, J. M., Micó, L., and Cardoso, J. S., editors, *Pattern Recognition and Image Analysis*, pages 133–140, Berlin, Heidelberg. Springer Berlin Heidelberg.

[3] Breuel, T. M. (2017). Robust, simple page segmentation using hybrid convolutional mdlstm networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 733–740.

[4] Chen, K., Seuret, M., Hennebert, J., and Ingold, R. (2017). Convolutional neural networks for page segmentation of historical document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 965–970.

[5] Chen, K., Wei, H., Hennebert, J., Ingold, R., and Liwicki, M. (2014). Page segmentation for historical handwritten document images using color and texture features. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 488–493.

[6] Cruz, F. and Terrades, O. R. (2014). Em-based layout analysis method for structured documents. In *2014 22nd International Conference on Pattern Recognition*, pages 315–320.

[7] Diem, M., Kleber, F., and Sablatnig, R. (2011). Text classification and document layout analysis of paper fragments. In *2011 International Conference on Document Analysis and Recognition*, pages 854–858.

[8] Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2017). A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, 64:1 – 14.

[9] Fawzi, A., Pastor, M., and Martínez-Hinarejos, C. D. (2017). Baseline detection on arabic handwritten documents. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, DocEng '17, pages 193–196, New York, NY, USA. ACM.

[10] Fernández, D., Lladós, J., Fornés, A., and Manmatha, R. (2012). On influence of line segmentation in efficient word segmentation in old manuscripts. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 763–768.

[11] Grüning, T., Leifert, G., Strauß, T., and Labahn, R. (2018). A two-stage method for text line detection in historical documents. *CoRR*, abs/1802.03345.

[12] Kesiman, M. W. A., Burie, J. C., and Ogier, J. M. (2016). A new scheme for text line and character segmentation from gray scale images of palm leaf manuscript. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 325–330.

[13] Kise, K. (2014). *Document Structure and Layout Analysis*, page 135. Springer London, London.

[14] Likforman-Sulem, L., Zahour, A., and Taconet, B. (2007). Text line segmentation of historical documents: a survey. *Int. J. Doc. Anal. Recognit.*, 9:123–138.

[15] Lu, Z., Schwartz, R., and Raphael, C. (2000). Script-independent, hmm-based text line finding for ocr. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 551 –554 vol.4.

[16] Meier, B., Stadelmann, T., Stampfli, J., Arnold, M., and Cieliebak, M. (2017). Fully convolutional neural networks for newspaper article segmentation. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 414–419.

[17] Messaoud, I., Amiri, H., Abed, H., and Margner, V. (2012). A multilevel text-line segmentation framework for handwritten historical documents. In *Proocedings of ICFHR*, pages 515–520.

[18] Moysset, B., Bluche, T., Knibbe, M., Mohamed Faouzi Benzeghiba, R. M., Louradour, J., and Kermorvant, C. (2014). The A2iA multi-lingual text recognition system at the second maurdor evaluation. In *Proocedings of ICFHR*, pages 297–302.

[19] Nagy, G. (2000). Twenty years of document image analysis in pami. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):38–62.

[20] Namboodiri, A. M. and Jain, A. K. (2007). *Document Structure and Layout Analysis*, pages 29–48. Springer London, London.

[21] Peng, X., Setlur, S., Govindaraju, V., and Sitaram, R. (2013). Handwritten text separation from annotated machine printed documents using markov random fields. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(1):1–16.

[22] Quirós, L. (2018). Multi-task handwritten document layout analysis. *CoRR*, abs/1806.08852.

[23] Schone, P., Hargraves, C., Morrey, J., Day, R., and Jacox, M. (2018). Neural text line segmentation of multilingual print and handwriting with recognition-based evaluation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 265–272.

[24] Song Mao, Azriel Rosenfeld, T. K. (2003). Document structure analysis algorithms: a literature survey.

[25] Stahlberg, F. and Vogel, S. (2015). Detecting dense foreground stripes in arabic handwriting for accurate baseline positioning. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 361–365.

[26] Surinta, O., Holtkamp, M., Karabaa, F., van Oosten, J.-P., Schomaker, L., and Wiering, M. (2014). A* path planning for line segmentation of handwritten documents. In *Proocedings of ICFHR*, pages 175–180.

[27] Wang, S., Baird, H., and An, C. (2009). Document content extraction using automatically discovered features. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1076–1080.

[28] Wei, H., Baechler, M., Slimane, F., and Ingold, R. (2013). Evaluation of svm, mlp and gmm classifiers for layout analysis of historical documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1220–1224.

[29] Wong, K. Y., Casey, R. G., and Wahl, F. M. (1982). Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656.

[30] Xu, Y., He, W., Yin, F., and Liu, C. (2017). Page segmentation for historical handwritten documents using fully convolutional networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 541–546.

CHAPTER 3

TEXT REGION DETECTION & CLASSIFICATION

Chapter Outline

3.1 Introduction

Handwritten Text Segmentation (HTS) is the task by which specific regions of text are detected and extracted. HTS can be considered one of the main tasks of DLA. Although HTS has an interest by itself for certain digitalization purposes, the most important uses of the results yielded are indirect. For example, *Text lines* are among the most interesting image regions that are required as input for certain higher order tasks, as we commented in Chapter 1.

HTS is not an easy task. Furthermore, region detection and classification in handwritten text entails a greater difficulty, in comparison with printed text. This is due to the inherent properties of handwritten text: variable inter-line spacing, overlapping and touching strokes of adjacent handwritten lines, bleed through, stains, ageing marks, etc.

As we commented in Chapter 2, HTS is currently applied at two different levels. Although this basic categorization mainly refers to "the order in which the information is processed" [6], it indicates the existence of two levels and makes the need for classification already apparent. At the most basic level, developing an approach that differentiates foreground elements from the background already adds the concept of classes. Regardless of these classes being considered in the method or not, they are present. Furthermore, the actual idea of having different levels of information and objective regions to be detected indicates to us that HTS is better tackled from the perspective of classification.

If this task is not tackled with the idea of classes at the heart of the method, some sort of constrain or restriction will appear. Text Region Detection and Classification (TRDC) at its various levels has been classically tackled via heuristic techniques, as we saw in Chap. 2. Unfortunately heuristic approaches are not really able to cope adequately with the high level of variability of page layout that real world documents present. Even worse, adapting to new region types as per requirements of the users or higher level systems is impossible with tailored methods.

Even inside the same information level, we might need to detect different region types as per a user requirement, which again adds to the idea of HTS being a classification problem. We could initially consider to just tackle the classification of the regions after the detection of all foreground elements has been performed, but that would force us to embed some knowledge of the *type* of foreground components we want to classify into our detection system.

This vicious cycle is surprisingly similar to the one observed in handwritten text recognition and key word spotting systems, where some methods attempt to segment the words in the text line image before recognizing each of them individually. Approaches which follow this idea accumulate errors coming from the sequential, decoupled combination of the detection and classification steps, generally leading to poor overall performance, both in word recognition and segmentation.

In the same manner this has been solved in modern handwritten text recognition and automatic speech recognition systems we intend to attack this issue in a holistic manner. By recognizing all the different elements in the page we will obtain the optimal region segmentation as a by-product.

This key idea will be the basis for our stochastic text region detection and classification (hereafter referred to as STRDC) framework.

Another issue with certain methods, which we will attempt to tackle, is trying to solve both the detection and extraction in a single strongly coupled method. This is specially the case in *Text Line Segmentation*(TLS) approaches, a subtask of HTS. In TLS, until recently, the *de facto* standard evaluation measure greatly disregarded the importance of the detection phase as we will see in Chapter 6.

Our proposed method breaks away from some of the established standards in the community (at the time the research for this thesis started). We recognized the importance in decoupling the detection and extraction subtasks, therefore we developed two independent techniques for each of them.

The page region detection subtask was tackled as a classification problem due to the aforementioned reasons. Our framework considers a page as a vertical stacking of different page regions. The developed approach can be applied to both levels of the HTS problem. By classifying each of the regions present in this vertical stack we obtain the actual segmentation coordinates as a by-product. To do this, our method applies tried-and-tested statistical modelling ideas to the HTS task. In Sec. 3.3 we will provide a detailed review of our detection and classification system.

Our region extraction technique was designed in order to build on the result yielded by the region detection and classification method. As we will see in Chapter 4 our method calculates an equidistant extraction frontier between two regions given their detection coordinates.

As indicated in Chapter 1 our stochastic framework makes some assumptions regarding the images it receives in order to work. First of all the framework assumes that the information will be received in a single-column area. This single-column area being composed of various roughly parallel horizontal regions that pertain to one of the classes being considered for the document.

This assumption is not unrealistic for the vast majority of handwritten documents of interest for transcription: large text image collections with a fairly homogeneous page structure. Additionally, the existence of methods [4, 15, 24] to perform column detection in a page image guarantees that our method can still be applied in the cases this assumption does not hold true.

It should be noted that, while the STRDC techniques here proposed can properly deal with mild variability of region slopes, large fluctuations are *not* properly supported[a]. Examples of the kind of line slope variability which is and is not supported are shown in Fig. 3.1.

[a]The proposed approach lends itself to extensions which would properly support the large slope drift shown in Fig. 3.1 (top), but these extensions are left for future work.

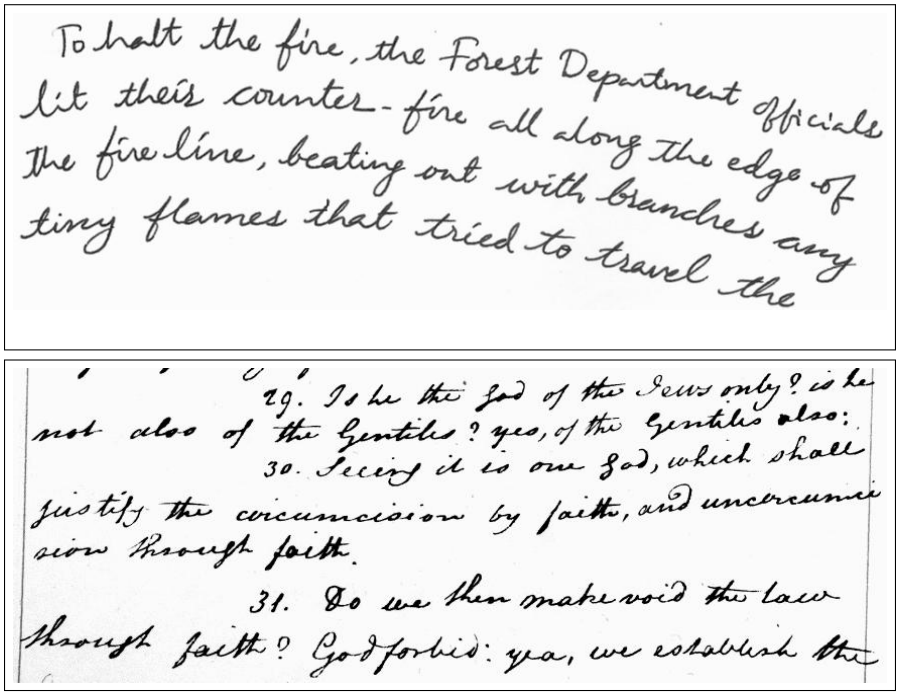


Figure 3.1: Examples of severe (top) and mild (bottom) line slope variability. The latter is supported by the proposed technique; the former is not.

In Fig. 3.1 we can also see a clear difference between the use of different line types in the documents. While the top sample image presents difficulty due to the text line warping the text lines are pretty similar. The bottom document is a clear example of a document where our approach clearly shines; The different text line types present makes our method an ideal fit for it.

3.2 Classical Approach to Text Line Detection

Before fully explaining our STRDC approach it is interesting to describe in detail a classical text line detection system. This method will help us in highlighting some of the issues with heuristic approaches and will also be used later in Chapter 7.

This method belongs to the Horizontal Projection Profile family of methods which we talked about in Chapter 2 and is mentioned in the various document layout technique surveys [6, 13, 16]. This family of techniques determines the text lines positions by analysing the peaks and valleys of the the full-page Horizontal Projection Profile (HPP).

These methods were widely used at different stages of document layout analysis and handwritten processing techniques [1, 7, 13, 18, 19, 25, 26]. Since our own approach uses these HPPs

as graphical features to train our stochastic based approach, as we will see in the next section, this allows us to showcase the qualitative difference between heuristic and machine learning based approaches.

We will now describe an image preprocessing procedure designed to enhance the foreground of our document images and aid the TRDC techniques in their estimations. Afterwards, we will provide detail on a classical HPP based technique that estimates the position of the text lines with the aid of the fast Fourier transform.

3.2.1 Classical Preprocess

The input data for any TRDC system are text images. As a first step, each image must undergo certain changes through the application of conventional *preprocessing* procedures.

Our base scanned input images of the handwritten documents require that its visual characteristics are improved/corrected in order to not impact adversely on the subsequent feature extraction and line detection estimation. This process is done due to the known issues of handwritten historical documents: low quality, stains and faint letters, loose formatting, narrow spaced lines, connected or overlapping components, and writing of the verso that appears on the recto due to bleed through.

The panels in Fig. 3.2 show, step by step, the images resulting from the successive application of preprocessing techniques to a portion of a page image. Next, we will provide full details of these techniques.

Partial background clean-up

We start from the original image (step 1 of Fig. 3.2) that contains stains, writing on the verso appearing on the recto and non-uniformity of the background colour which makes it difficult to process. In order to eliminate these issues we first perform a grey-level normalization (step 2 of Fig. 3.2) and apply to the resulting image a bi-dimensional median filter [12]; we subtract the result of the filter from the original image and obtain the result displayed in step 3 of Fig. 3.2.

Skew correction and final clean-up

Once the background is partially cleaned we can proceed to correct the skew. Skew is usually a distortion introduced during the document scanning process but can also be an issue introduced by the writer. Skew is understood as the angle of the document paper with respect to the scanner coordinates system. Skew must therefore be corrected one page image at a time, by aligning the text lines present with the horizontal axis.

Skew correction is carried out by searching for the angle which maximizes the variance of the horizontal projection profile and then applying a rotation operation with the calculated angle [10].

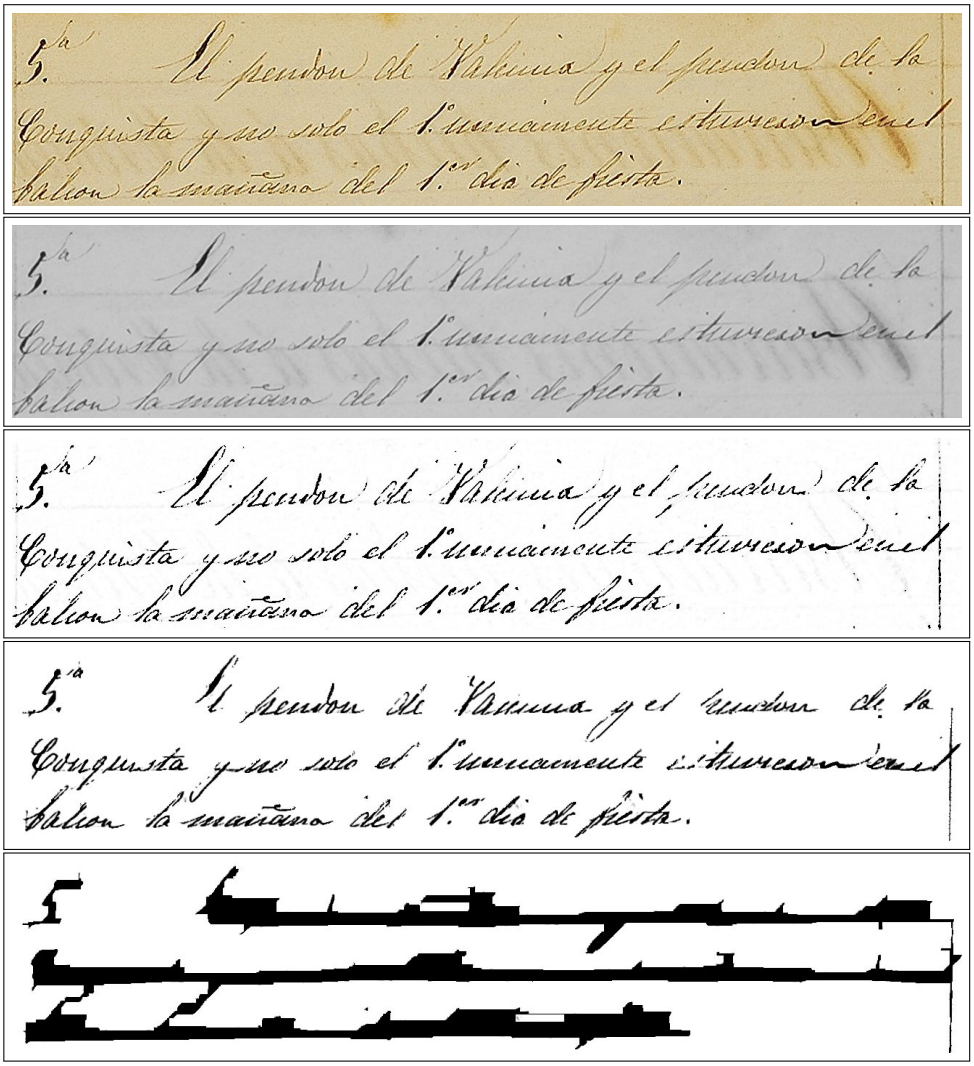


Figure 3.2: Illustration of preprocessing steps on a fragment of a handwritten text image.

We first execute the run length smear algorithm (RLSA) [32] to enhance the horizontal projection profile (step 4 of Fig. 3.2) and then we calculate the angle which we use to correct the skew by applying a rotation operation.

Once the image is finally deskewed we perform a final clean-up by executing a global image filter inspired in the Suavola method [31]. This technique does not binarize the image and enhances greatly any faint groundtruth stroke. With this we obtain the final clean and skew corrected result that can be seen in step 4 of Fig. 3.2.

Text smearing

Finally *run length smearing algorithm* (RLSA) [32] is performed again. This smearing of the deskewed and cleaned foreground text regions, is performed with the objective to emphasize the resulting horizontal projection profiles (HPP). More information on how to calculate these profiles will be provided in Sec. 3.3.1. It is important to note that until this point we were working

with a greyscale image. We improved the image by eliminating all noise and fleshing out all real foreground text strokes, so that when the RLSA was executed, all the foreground regions would be positively impacted.

3.2.2 HPP: Peak and Valley Estimation

The resulting HPP is analysed to find significant maxima (or peaks) and minima (or valleys) according to max-min thresholds which need to be heuristically tuned for the text images considered.

One of the most serious drawbacks of the traditional HPP-based approach is that short and narrow text lines often fail to produce significant HPP peaks. To overcome this problem an overall interline separation is estimated by means of Fourier Analysis as the period of the highest energy harmonic. This estimate is more robust than other popular estimates obtained (for instance) by averaging vertical distances of consecutive HPP valleys.

The estimated interline distance allows us to predict potential line detection gaps; i.e., vertical regions where the previous analysis failed to find any significant HPP peak and valley. These gaps can then be reviewed using lower, adaptive thresholds, thereby leading to improved overall detection performance.

This extra information allows to outperform more traditional approaches, by improving detection of short lines and rejecting those whose estimated HPP heights do not to reach the predefined threshold. For example, this improvement allows us to better detect small intercalated text between text lines.

Even if it is obviously not perfect, this technique was proven to be robust and accurate enough at the time. It has been used reliably for detecting text lines needed in many works of handwritten text recognition and related tasks [23, 27].

Although this classical approach provided some adequate results, it does present certain issues: detection of very different text line types, application to large region types or fine tuning the threshold parameters for corpus with mild variability.

Next we will review in detail our STDC approach. Although it uses the same HPP for feature extraction it is able to resolve the aforementioned issues.

3.3 Stochastic Text Region Detection and Classification

As we stated in the introduction, our approach will follow the successful statistical modelling ideas used in automatic speech recognition (ASR) and handwritten text recognition (HTR). In statistical language processing:

- Low-level elements, such as phonemes in ASR, or characters in HTR, are usually modelled by hidden Markov models or neural networks.
- Characters and words are linked at an intermediate level by means of a lexicon.
- Stochastic finite-state networks (N-gram models) are typically used to model the constraints which rule the concatenation of words into the highest-level entities (usually sentences) [11].

In our STRDC task, the low-level elements will be called "text *layout elements*" (LE), while the intermediate-level is composed of "text *layout regions*" (LR), which can be vertically stacked ("concatenated") to form text blocks or pages depending on the kind of regions the system must tackle.

For example, if the task is to detect larger region types such as paragraphs, title lines, lyric section, diagrams, etc., the layout elements will be larger graphical sections of a page (like the start, middle and ending of a paragraph) that allows us to better identify the layout regions. These layout regions will be composed in order to form a whole text block. If smaller regions like text lines are being considered, the layout elements might be much smaller graphical components of a page; like the interline space between two adjacent text lines.

Defining the adequate LEs and LRs depends on the requirements of the TRDC task and/or the characteristics of the documents considered. As we will see in the next sections, the framework we are presenting in this thesis provides the user easy ways to add this information and modelling. We will see many examples of the layout element and region modelling being applied to different corpus in Chap. 5.

The *vertical shapes* of LEs are described by stochastic ("optical") *line element models*. To be precise, in our approach we will be using Hidden Markov Models to perform this task.

The intermediate structure which links LRs and LEs is called *"LR dictionary"*. As with the lexicon in ASR or HTR, this dictionary is used here to deterministically specify how to form LRs by concatenating elements from the LE "alphabet". Finally, one should take into account that LRs can *not* be stacked arbitrarily; for instance, after a start-paragraph line it is very unlikely to encounter another start-paragraph line or a header-line; or after a regular line, it is most probable to find another regular line. Similarly, physical page dimensions and script size impose restrictions on the number of lines which may appear in a page. This kind of highest-level constraints are modelled by a a stochastic finite-state (or N-gram) *vertical layout model* (VLM), which plays the same role as language models do in typical language processing tasks.

As will be discussed in Sec. 3.3.2 and in Sec. 3.3.3 both the line element models and the vertical layout model can be easily trained from lightly annotated training page images. The ground-truth needed to train both the vertical shape HMMs and the VLM is extremely simple: It just consists in a sequence of labels qualitatively describing the successive LRs (or LEs) which appear in the page.

Once these models are available, TRDC of new, unlabelled handwritten text images can be performed by means of Viterbi decoding [11]. As will be discussed in Sec. 3.3.2, this provides for each page image an optimal sequence of LR (and LE) labels, as well as the corresponding optimally estimated vertical position of each region in the page.

Fig. 3.3 shows a diagram of the proposed STRDC approach, which entails four main steps: image preprocessing, feature extraction, training and decoding.

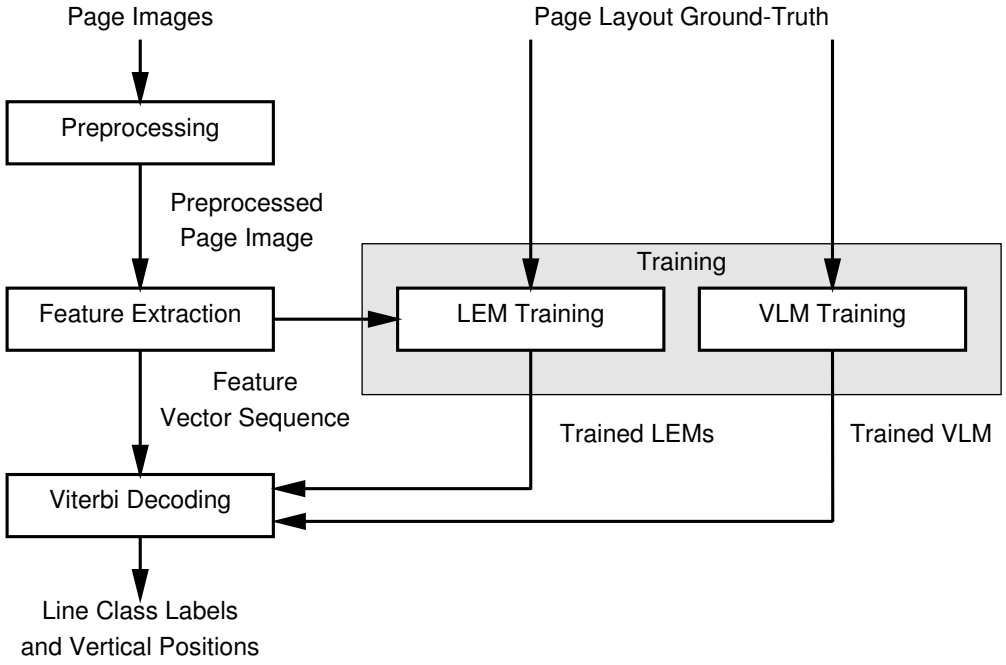


Figure 3.3: A diagram of the proposed STRDC approach. No region geometric position information is needed in the training data. LEM and VLM stands for "layout element model" and "vertical layout model", respectively.

The *Preprocessing* phase consists in the application of the image cleaning techniques explained in Sec. 3.2.1. Once the image is cleaned the information present in them is condensed into graphical features. In Sec. 3.3.1 we will explain how this *Feature Extraction* is performed.

Concerning the *Training* phase, we mean the process carried out to estimate the optimal parameter values of the involved models, i.e. HMMs and VLM (Sec. 3.3.3), from available training samples. Finally, the *Decoding* phase comprehends the process that takes as input a page image (represented as a sequence of feature vectors) and outputs the corresponding sequence of detected line classes along with their placement information (Sec. 3.3.2).

We will now proceed to explain the *Feature Extraction* calculation phase in detail.

3.3.1 Feature Extraction

As our STRDC approach is based on Hidden Markov Models (HMMs), this implies that we must present the page image as a sequence of feature vectors. Initially, we could simply just consider the values of each cell along a pixel row in the image as our feature vector.

Such a naive feature vector would impact negatively the training and decoding of the of the HMMs. From an operative perspective, the larger the feature vector the longer it would take to train and decode the models. Furthermore, the accuracy could be impacted by an excess of features: they could create noise instead of actually making the discrimination between classes easier. Additionally, providing the information present in the page in a synthetic manner will allow the models to better grasp contextual knowledge.

Due to the aforementioned reasons, we must choose adequately what features we add to our vectorial representation of each page. We must ensure that all the relevant information that could aid our models to classify correctly the page regions is present, but we must also seek to provide it in the most condensed manner in order to not degrade training and decoding performance.

This is performed in the following manner. The original image I, of height L and width M, is preprocessed as per the method described in Sec. 3.2.1. The preprocessed image X is divided into D non-overlapping rectangular regions, each of them with the same height of the whole image, L.

In each of these regions $d : 1 \leq d \leq D$, the *horizontal projection profile* (HPP) [16] is computed by scanning it row-wise along the horizontal axis and counting the number of black pixels in each row $l : 1 \leq l \leq L$ with the following function $\mathcal{H}$:

$$\mathcal{H}(X, d, l) = \frac{\sum_{j=(d-1)\cdot m+1}^{d\cdot m} X_{lj}}{\sum_{k=1}^{L} \sum_{j=(d-1)\cdot m+1}^{d\cdot m} X_{kj}} \tag{3.1}$$

Where m is the width of each of the D rectangular regions calculated easily as $m = \frac{M}{D}$.

The resulting HPP values are then smoothed through a rolling average filter [17]. Examples of the smoothed HPPs for $D = 5$ can be seen in Fig. 3.4 where we can see how they are able to represent adequately where the foreground text is present in the image.

From the HPPs, a D-dimensional feature vector is obtained for each row of pixels where each component of the vector is the corresponding projection profile value for that region. Hence, at the end of this HPP calculation, a sequence of L D-dimensional feature vectors is obtained.

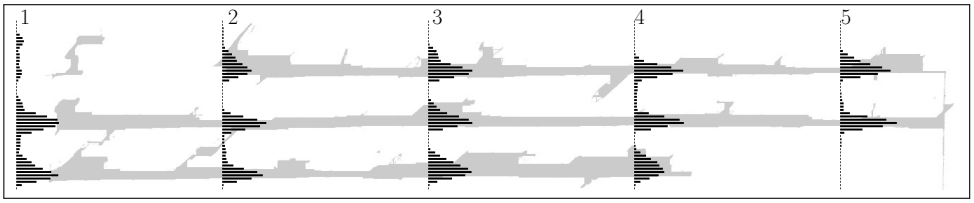


Figure 3.4: Illustration of the resulting smoothed horizontal projection profiles overlayed on top of the image portion it was calculated on.

Borrowed from ASR and HTR, feature vectors are augmented by including HPP derivatives, calculated according to [33]. As it can seen in Fig. 3.5, derivatives provide more explicit information about the transition between white space, ascenders, main line body and descenders.

At the end of this graphical feature extraction process, for an image of height L divided into D rectangular boxes, a feature vector sequence, $\boldsymbol{o} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_L$, is obtained where, for each i, $\boldsymbol{o}_i \in \mathbb{R}^{2D}$.

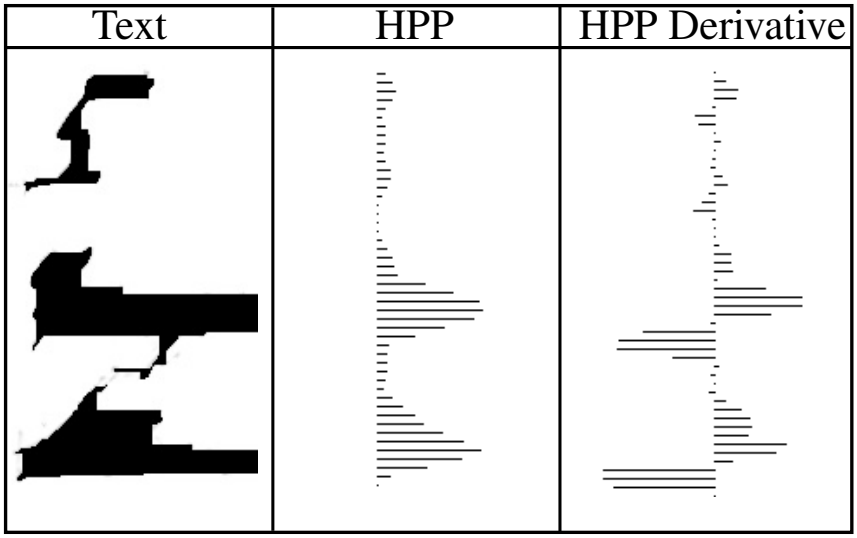


Figure 3.5: A fragment of a rectangular text region and its corresponding HPP and HPP-derivative features.

3.3.2 Decoding

Next, we will describe in a more formal manner the decoding phase of our STRDC approach. The decoding phase will describe, parting from the theoretical background provided in Sec. 1.3, how we apply the HMM and VLM statistical models to the TRDC task.

As previously commented, we formulate STRDC as the problem of finding a most likely LR label hypothesis sequence, $\hat{\boldsymbol{h}} = \hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n$, for a given handwritten page image (or selected

image block) represented by an observation sequence[b] $\boldsymbol{o} = \boldsymbol{o}_1^L = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_L$, that is:

$$\hat{\boldsymbol{h}} = \arg\max_{\boldsymbol{h}} P(\boldsymbol{h} \mid \boldsymbol{o}) \tag{3.2}$$

Using the Bayes' rule this can be rewritten as:

$$\hat{\boldsymbol{h}} = \arg\max_{\boldsymbol{h}} P(\boldsymbol{h})\, P(\boldsymbol{o} \mid \boldsymbol{h}) \tag{3.3}$$

where $P(\boldsymbol{o} \mid \boldsymbol{h})$ and $P(\boldsymbol{h})$ are, respectively, the optical *line element model* (LEM) and the *vertical layout model* (VLM). Both these models were introduced during the beginning of this STRDC section (see page 39) and will be covered in more detail in Sec. 3.3.3.

In TRDC, we are interested not only in adequately labelling the LRs of a given text image, but also in determining the vertical positions of the corresponding lines in the image. In STRDC, this is similar to finding an optimal segmentation of a speech signal into word segments associated with an optimally decoded sequence of words. Formally speaking, LR vertical positions are implicit or "hidden" in $p(\boldsymbol{o} \mid \boldsymbol{h})$. To explicitly uncover them, Eq. (3.3) can be rewritten as:

$$\hat{\boldsymbol{h}} = \arg\max_{\boldsymbol{h}} \; P(\boldsymbol{h}) \sum_{\boldsymbol{b}} P(\boldsymbol{o}, \boldsymbol{b} \mid \boldsymbol{h}) \tag{3.4}$$

where $\boldsymbol{b}$ is a *segmentation*; that is, a sequence of $n+1$ *boundary marks*, $b_0, b_1, \ldots, b_n$, such that $b_0 = 0,\ b_i < b_j, 0 < i < j < n,\ b_n = L$.

Now, if we approximate the sum in Eq. (3.4) with the dominating term being added we obtain:

$$\hat{\boldsymbol{h}} \approx \arg\max_{\boldsymbol{h}} \; P(\boldsymbol{h}) \max_{\mathbf{b}} p(\mathbf{o}, \mathbf{b} \mid \mathbf{h}) \tag{3.5}$$

from which we can arrive to the following formula:

$$(\hat{\boldsymbol{b}}, \hat{\boldsymbol{h}}) \approx \arg\max_{\boldsymbol{b},\boldsymbol{h}} P(\boldsymbol{h})\, P(\boldsymbol{o}, \boldsymbol{b} \mid \boldsymbol{h}) \tag{3.6}$$

where $\hat{\boldsymbol{b}}$ is an optimal segmentation as per our approximation. The pair $(\boldsymbol{o}, \boldsymbol{b})$ in the above equations actually represents a sequence of feature vector sub-sequences, $\boldsymbol{o}_{b_0+1}^{b_1}, \ldots, \boldsymbol{o}_{b_{n-1}+1}^{b_n}$. Therefore, the chain rule can be applied to expand Eq. (3.4) as:

$$\begin{aligned}(\hat{\boldsymbol{b}}, \hat{\boldsymbol{h}}) = \arg\max_{\boldsymbol{b},\boldsymbol{h}} P(\boldsymbol{h})\; & P(\boldsymbol{o}_{b_0+1}^{b_1} \mid \boldsymbol{h})\, P(\boldsymbol{o}_{b_1+1}^{b_2} \mid \boldsymbol{o}_{b_0+1}^{b_1}, \boldsymbol{h}) \\ & \ldots\; P(\boldsymbol{o}_{b_{n-1}+1}^{b_n} \mid \boldsymbol{o}_{b_0+1}^{b_1}, \ldots, \boldsymbol{o}_{b_{n-2}+1}^{b_{n-1}}, \boldsymbol{h})\end{aligned} \tag{3.7}$$

[b] For a sequence $\boldsymbol{o}$, $\boldsymbol{o}_i^j$ denotes the subsequence $\boldsymbol{o}_i, \ldots, \boldsymbol{o}_j,\ \ i \leq j$.

Finally, since $\boldsymbol{h} = h_1, \ldots, h_n$, we can (reasonably) assume that, given h_i, the subsequence $\boldsymbol{o}_{b_{i-1}+1}^{b_i}$ is independent of all $h_j, j \neq i$, and also of all the previous feature sub-sequences[c] $\boldsymbol{o}_{b_0+1}^{b_1}, \ldots, \boldsymbol{o}_{b_{i-2}+1}^{b_{i-1}}$. This yields:

$$(\hat{\boldsymbol{b}}, \hat{\boldsymbol{h}}) \approx \arg\max_{\boldsymbol{b},\boldsymbol{h}} P(\boldsymbol{h})\, P(\boldsymbol{o}_{b_0}^{b_1} \mid h_1) \ldots P(\boldsymbol{o}_{b_{n-1}}^{b_n} \mid h_n) \tag{3.8}$$

which exactly corresponds to the optimization solved by the Viterbi search algorithm [11]. Since LRs are specified as deterministic concatenations of LEs, we can apply Viterbi decoding, described in detail in Sec. 1.3.1. Viterbi straightforwardly provides not only the LR boundary marks $\hat{\boldsymbol{b}}$, but also the more fine-grained boundaries between the corresponding LEs.

In practice log-probabilities are used and, as happens in ASR and HTR, coefficients are added to the equation to perform further fine tuning. A coefficient called *"VLM scale factor"*, α in our formulation, is used to affect the $\log P(\boldsymbol{h})$ term. A length-dependent term βn is added, where β is called *"LR insertion penalty"* and n is the length of $\boldsymbol{h}$. Both α and β are tuned empirically to balance the contribution of the two kinds of models involved in Eq. (3.3) [11]. By adding these two factors to Eq. (3.8) we end up with the final decoding equation Eq. (3.10).

$$(\hat{\boldsymbol{b}}, \hat{\boldsymbol{h}}) \approx \arg\max_{\boldsymbol{b},\boldsymbol{h}} \frac{P(\boldsymbol{h})^{\alpha}}{n^{\beta}}\, P(\boldsymbol{o}_{b_0}^{b_1} \mid h_1) \ldots P(\boldsymbol{o}_{b_{n-1}}^{b_n} \mid h_n) \tag{3.9}$$

$$(\hat{\boldsymbol{b}}, \hat{\boldsymbol{h}}) \approx \arg\max_{\boldsymbol{b},\boldsymbol{h}} \sum_{i=1}^{n} \log P(\boldsymbol{o}_{b_{i-1}}^{b_i} \mid h_i) \; + \alpha \log P(\boldsymbol{h}) - \beta n \tag{3.10}$$

3.3.3 Modelling and Training

As we stated in the introduction, our system will follow the successful statistical modelling ideas used in automatic speech recognition (ASR) and handwritten text recognition (HTR). More specifically, in our framework the modelling will be performed as follows.

- The Low-level elements that correspond to sub areas of the document, denominated as *Layout Elements*(LE) in our framework, will be modelled by hidden Markov models.
- The actual *Layout Regions* (LR) we are searching are the intermediate level. The LRs will be formed by composition of the different LEs by means of a lexicon.
- A Stochastic finite-state automata model (N-gram model) will be used to model the constraints which rule the concatenation of the different LRs in order to correctly form a document page. [11].As the page is seen as composed of different *vertically* concatenated LRs, we denominate the language model that restricts the composition the *Vertical Layout Model* (VLM).

[c] See output independence assumption Sec. 1.3

These three modelling levels have already been introduced in Sec. 3.3.2. The models actually correspond to the different terms of our decoding equation Eq. (3.10). We will now proceed to explain in detail each of the modelling levels, what they represent and how they are trained.

Optical Models - Layout Elements:

The term $p(\boldsymbol{o}, \boldsymbol{b} \mid \boldsymbol{h})$ in Eq. (3.6) is obtained as the product of the individual likelihoods of the successive segments of $\boldsymbol{o}$, given the corresponding successive labels of $\boldsymbol{h}$, $h_1, \ldots, h_n$. Initially, the labels $\boldsymbol{h}$ could be considered to correspond to each of the *Layout Regions* we require to detect and classify. In reality, these LR themselves are composed by stacking simpler elements we will name as *Layout Elements*[d].

For example a Paragraph region could be formed by a beginning-of-paragraph element, a mid-paragraph element and an end-of-paragraph element. Hence, this term refers to each of the LEs probabilistic models, the lowest-level of modelling in our STRDC approach. In this work, we have chosen to model LEs by continuous density left-to-right hidden Markov models (HMM)[e].

In Fig. 3.6 we can see an illustration of an HMM being used to detect/train an specific LE. In this case our STRDC is being used to detect and classify different lines (our LRs): short lines, normal lines, paragraph lines. To do so we define the different LEs that via composition allow us to represent these LRs. In this case we define Blank Space (BS), Line Body (LB), Non-text (NT), Short Line (SL), Paragraph Line (PL) and Interline Space (IL). Some of these LEs (LB,NT,SL,PL) have a direct translation to the LRs and the task requirements while others are required for accuracy purposes (BS,IL). Defining the adequate LRs and LEs for the specific corpus is up to the designer as we will see in Chapter 5.

The adequate number of HMM states and Gaussians per state for a LE is determined empirically and may be conditioned by the amount of training data available for a specific corpus. Once an HMM "topology" (number of states, structure and number of mixture components) has been adopted, the HMM parameters can be easily and fully automatically trained.

The training data required for each page is the feature vector set for the specific page ($\boldsymbol{o}$, in Eqs. (3.3-3.6)), accompanied with a simple list of LR labels (which will be automatically decomposed to the relevant LEs). This training process is carried out using the well known instance of the EM algorithm called forward-backward or Baum-Welch re-estimation [11] which we reviewed in Sec. 1.3.1.

It is very important to note that this estimation technique does not need any sort of information about the actual positions, heights or boundaries of each of the LRs/LEs in the handwritten text image. This is a clear advantage over other statistical models being used for the optical modelling

[d]How the different *Layout Elements* stack to form the *Layout Regions* is defined in the LR dictionary that will be the next model we cover (Page 47).

[e]Introduced in this thesis in Sec. 1.3.

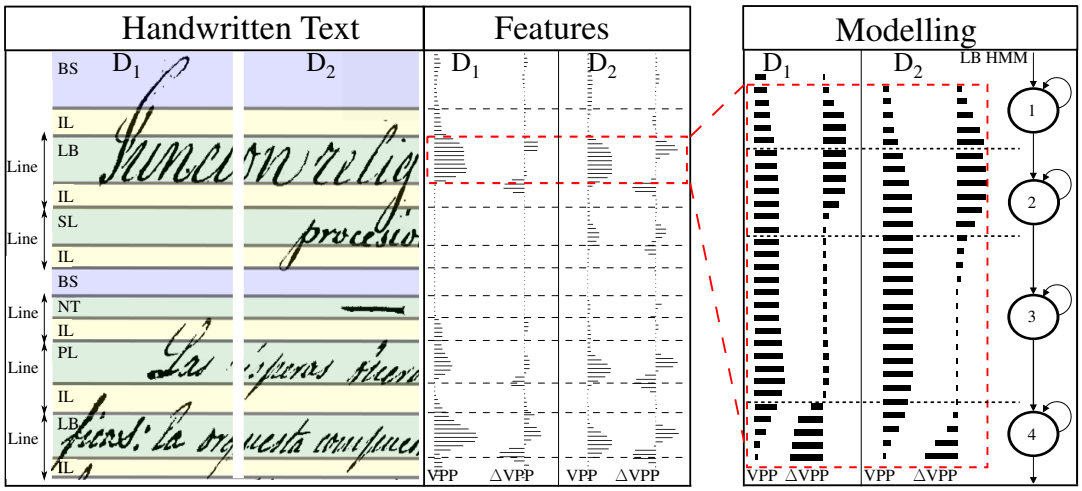


Figure 3.6: Examples of basic line element (LE) regions (`BS`, `IL`, `LB`, `SL`, etc.), along with a corresponding feature vector sequence. The right panel shows how a HMM LE model (for the LE "`LB`") is expected to model (and implicitly segment) this sequence.

task. As we will see through out our experimentation (Chap. 7) this reduction in the required training information will prove to be very convenient and is something to be considered if an approach is to be used in actual production scenarios (Sec. 7.6).

Lexicon Model - Layout Region Dictionary

As mentioned in the optical models section above, the term $p(\boldsymbol{o}, \boldsymbol{b} \mid \boldsymbol{h})$ relates the likelihood of the sequence of observations $\boldsymbol{o}$ to the sequence of labels $\boldsymbol{h}$. The set of accepted labels correspond to the type of *Layout Regions* that we are trying to detect and classify.

The *Layout Regions* are in turn composed by the vertical stacking of various *Layout Elements*. The intermediate structure that links the LRs and LEs is called the *"Layout Region Dictionary"*.

The dictionary can serve as a tool to deterministically specify certain knowledge we have regarding basic page structure and handwritten text.

For example, when applying STRDC to text line detection, we might need for a specific element to denote the interline space (IL) that can be found after a conventional text line (TL). Since the IL can only be present after a TL we can implement this restriction in the dictionary by defining the LR text line (LINE) as the concatenation of a TL element and an IL element:

```
LINE TL IL
```

We can also use the dictionary to indicate that a specific LR can be formed by two different LE sequences. If we consider for example a paragraph detection task, the main LR to consider would be the actual paragraph (PAR). The text paragraph region has two very well defined graphical

elements the start paragraph area (B) and the end paragraph area (E). Additionally, depending if the paragraph is very long or not we might need an additional element to model the mid section of text (M) between the beginning and end of a paragraph. In this case this would lead two the following two rules to be present in the dictionary:

```
PAR B E
PAR B M E
```

Letting the decoder decide which of the rules to apply in order to decode optimally the observation sequence as per the parameters (α and β) and the trained optical models.

Language Models - Vertical Layout Models

The higher level structure denominated *Vertical Layout Model* (VLM) corresponds to the $P(\boldsymbol{h})$ in Eq. (3.6). This model describes, in a probabilistic manner, how the different LRs are likely to follow each-other in order to compose a well formed text page. This term, $P(\boldsymbol{h})$, can be decomposed in an exact manner as follows:

$$P(\boldsymbol{h}) = P(h_1)P(h_2 \mid h_1) \dots P(h_n \mid h_1, \dots, h_{n-1})$$

This exact decomposition makes it clear that estimating the probabilities of the long-term dependencies described in it is not feasible in practice. Thus, we must cluster these dependencies into common *"histories"* in order to reduce the total number of probabilities to be estimated by which we make the whole computation affordable.

The resulting approximation to $P(\boldsymbol{h})$ can be described by a *stochastic finite-state model* [30] and, as a particular case, by a conventional N*-gram* model where common histories correspond to short-term dependencies (of length $N-1$); that is:

$$P(\boldsymbol{h}) = P(h_1)P(h_2 \mid h_1) \dots P(h_n \mid h_{n-N+1}, \dots, h_{n-1})$$

The parameters of this resulting model can be easily estimated by maximum likelihood by just computing N-gram frequencies of occurrence on a training set of LR sequences. A more detailed account of N*-gram* model definition and its estimation was covered in Sec. 1.3.2.

The VLM level greatly differentiates our STRDC framework from other, both heuristic and machine learning based, approaches. The VLM allows us to easily add prior probability information about what is expected as correct Layout Analysis result. Furthermore, through the VLM we can easily impose additional (deterministic) restrictions to include expert knowledge regarding the collection.

For example, information regarding the minimum and/or maximum number of LRs to be detected, record-type structures with certain variability, item enumerations, etc... which can greatly improve the accuracy of a result can easily be added via the VLM.

In order to facilitate the inclusion of the information regarding the known layout regularities, VLMs are implemented as stochastic finite state grammars (SFSG). SFSGs can properly represent N-gram probabilities [30][f] while making the manual addition of finite-state structures easy.

Since both the VLM and the LE HMMs are essentially finite-state models, they can be straightforwardly integrated into a single global finite-state model. Solving Eq. (3.8) or Eq. (3.10) can easily be achieved in this integrated model. That is, given an input page image represented as a sequence of feature vectors, an optimal output string of recognized line regions labels, along with the corresponding vertical position coordinates, is obtained.

Throughout the different experiments presented in Chp. 7 we will see the significant and positive impact the VLM has on the detection and classification process. We will review the effect of increasingly restrictive VLMs, inclusion of layout regularities and the use of VLMs in production scenarios.

3.4 Semi-automatic Iterative Production Process

Probabilistic approaches, like our own, have been proven to yield similar or better results than other traditional solutions. Yet, they are not often adopted for real use due to the (*incorrect*) belief that these systems always require large quantities of manually-labelled training data.

It is our duty to not only provide new approaches or technologies but also to describe how to optimally apply them in real production scenarios. Accordingly, we defined a semi-automatic iterative process for STRDC in order to tackle this usability issue.

Our production approach makes use of our proposed STRDC system (Sec. 3.3), along with user supervision, in order to yield regions of ground-truth quality in historical handwritten documents.

We will now describe the process. To start, a small amount of pages is manually labelled by a user or operator. Depending on the types of regions that we are trying to detect or classify, the labelling will be more or less complex. In our case, this labelling task will be always rather simple. The worst-case scenario would be a classification task that would just require a simple label list of the regions found on each training page. The easiest would be, for example, a line detection task for which our approach would only need the number of lines present in each page. Using the provided information, the TRDC models are trained.

Since the VLM is fixed for the specific task, training consists in performing a Baum-Welch estimation [11] of the HMM parameters from the feature vector sequence of the training pages, along with the corresponding *layout region* label sequences. This simplicity in the data requirement needs is due to the fact that the EM algorithm used for training doest not require the hidden

[f] Also covered in Sec. 1.3.2

variables to be provided. Hence, the STRDC system can train its models without the need of any geometric information.

Then, the system provides an initial hypothesis for the next batch of text baselines, which are revised (and corrected if needed) by the operator and added to the page layout ground-truth.

Finally, the TLAD system is re-trained using this ground-truth before processing the next batch of page images. A flow diagram of this iterative semi-supervised process for base line detection is shown in Fig. 3.7.

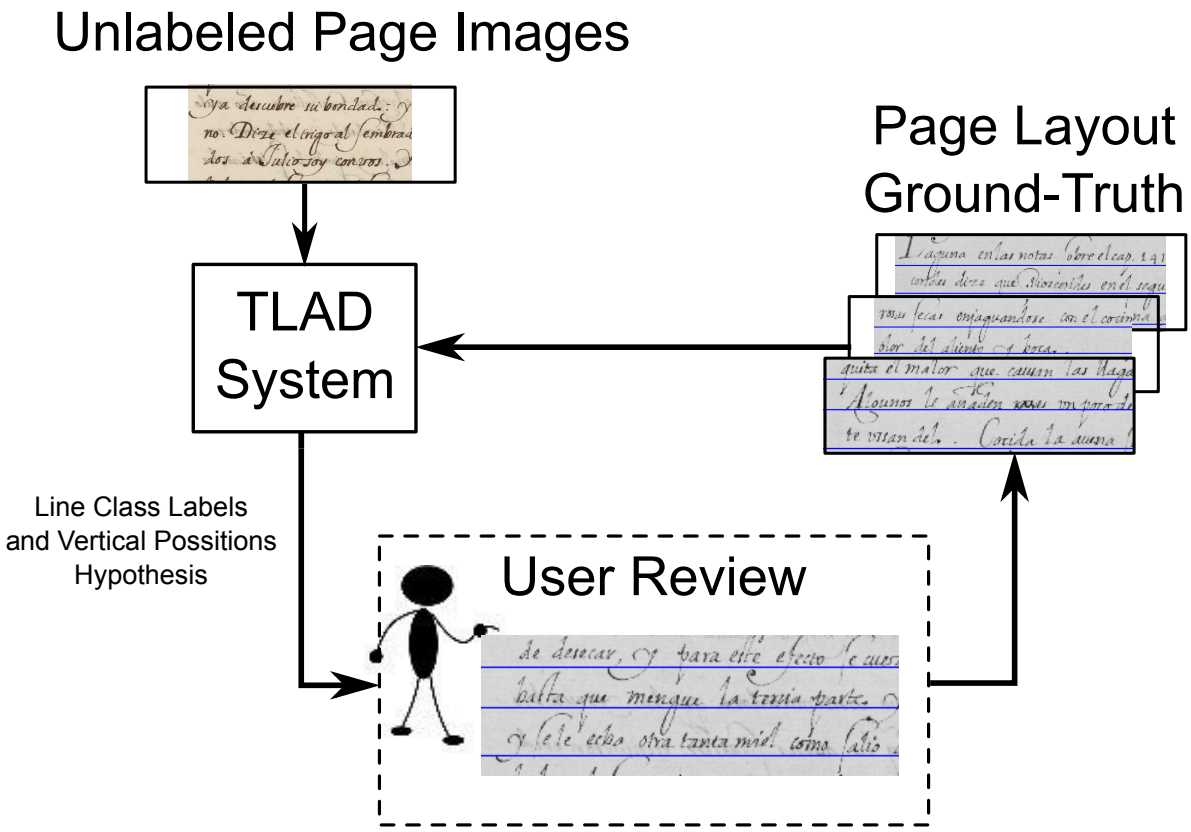


Figure 3.7: A flow diagram of the proposed iterative process for semi-automatic line detection and ground-truth creation.

The defined process not only yields ground-truth quality vertical layout information, but also significantly reduces the required operator's work load, which now only consists of revising (hopefully good) system-provided hypotheses. Furthermore, as more data is available for training the quality of the hypothesis should improve hence gradually reducing the human review process over time.

Later, in Sec. 7.6, we will review this process in baseline detection experiments with real users to evaluate our claims in real production scenarios.

3.5 Tandem: Convolutional Neural Networks and Hidden Markov Models

Currently, there is a renewed interest in Deep Learning (term introduced in 1986 [5]) and in the application of Deep Neural Networks (DNN) to solving different tasks. At the time the research work for this thesis started the usage of DNN in its different flavours had not taken off and it is only recently that we are starting to see publications of DNN being applied to HTS.

If we look at the most current works about convolutional neural networks (CNN) [20, 22] applied to the TRDC task, we can observe that these networks are often used to produce "*simplified images*". These simplified images eliminate all types of noise present in the image and mark areas where the desired regions are present with some likelihood. It is important to note that actual decision of how many regions are present, their localization and vectorization are left to "*post-processing*" tasks.

If we observe the final result of applying CNNs to an image with the idea of detecting text lines, the yielded image is rather similar to the final output of our preprocessing phase 3.2.1. This can be observed in Fig. 3.8.

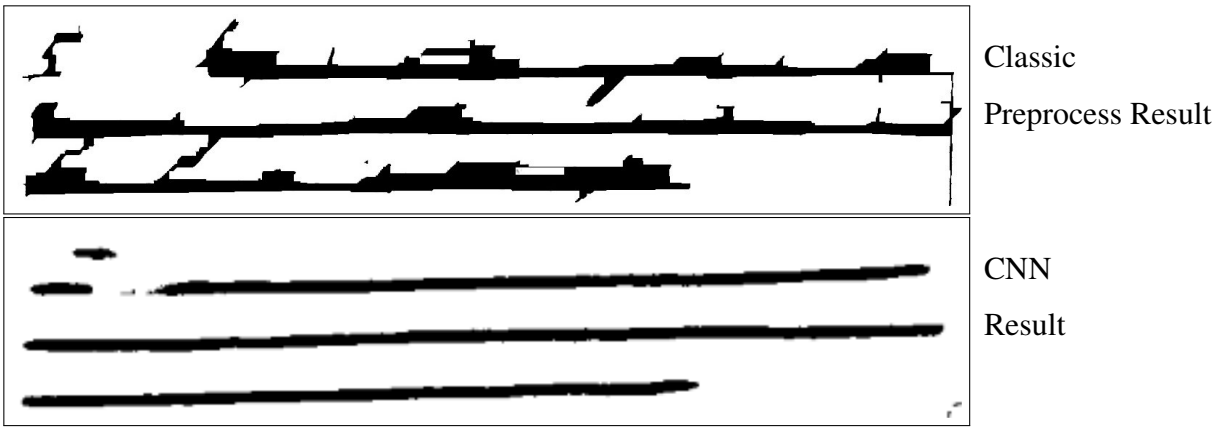


Figure 3.8: Illustration of preprocessing steps on a fragment of a handwritten text image.

Of course, the results obtained by means of the CNNs are of higher quality than the ones obtained by means of classical preprocessing techniques. Furthermore, they are obtained in an automatic manner without the need of manual fine tuning of parameters for the corpus. This is due to the training with labelled data that CNNs undergo. It is important to note, that regardless of the origin of the final image (preprocessed by composition of heuristic image tools or generated by a CNN) the actual decision of where the page regions of interest are located is still to be made.

With this in mind, the statistical HTS approach being presented in this thesis can still be used, as a tandem, to help determine the actual location and class of the regions. The CNN's resulting image can be used as the input image to calculate the feature vectors for the HMMs. Our STRDC

approach would add the vertical restrictions and take profit of the layout regularities, thus improving the base result provided by the CNN. This tandem approach has also been applied in ASR [9] and HTR [3] with positive results.

3.6 Text Content Based Features

In the last sections we have have covered the basics of our STRDC approach. We also reviewed how to effectively deploy the developed technique for production purposes and the benefits of combining with other probabilistic models like CNNs. Throughout all of this we only considered graphical information to detect and classify the target regions. Unfortunately, there are layout analysis problems where performing this task only using graphical information is impossible.

In many cases this happens because the considered document does not actually exhibit any graphical clue to distinguish between regions. Sometimes there are some sort of graphical clues to do so but they are not consistent. Other times the region type might only be known if we consider the graphical information of the following page. In Fig. 3.9 we see a sample page that exemplifies this issues. The page contains three records: There is no graphical gap between any adjacent records, the capital letter that marks the beginning of the second record is not present at the beginning of the third record, there is no graphical way to distinguish if the third record finishes in this page or continues in the next.

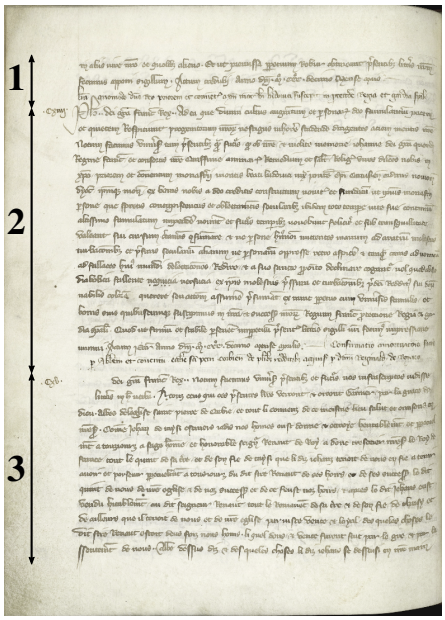


Figure 3.9: Sample Chancery corpus (Sec. 5.10) page that exemplifies the issues of depending solely on graphical features to distinguish between 3 regions.

To tackle this situation we study the (additional) use of text content based features to improve the accuracy of our HTS approach. Since layout analysis has traditionally been considered a step that must be performed previously to any sort of text recognition, wanting to use text content based features generates a vicious circle: text recognition requires previous layout analysis to be performed, but in its turn, some sort of text recognition is required to obtain these textual content features required for the layout analysis.

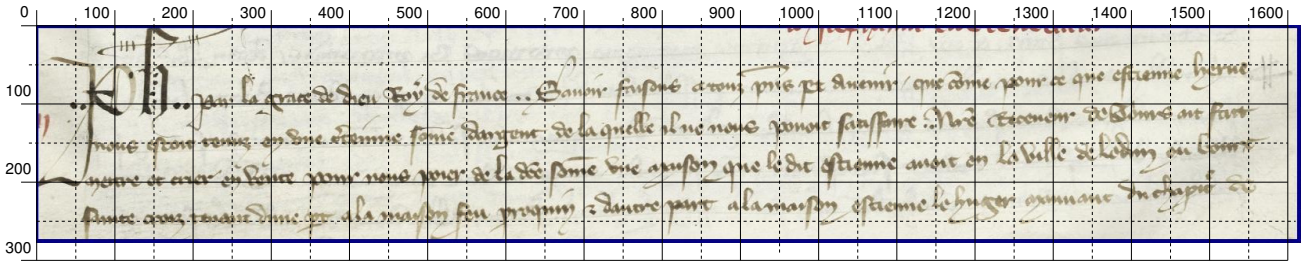
One way to obtain this kind of content-based features could be based on some sort of human labelling on the contents of each area page. An even more unrealistic requirement would be to have the transcribed text and coordinates of where each word is positioned. Obviously, the need of the transcribed text of a page in order to perform its document layout analysis of it would defeat the purpose altogether.

So that we may circumvent this deadlock, in a realistic manner, we rely on the recently introduced concept of "*Probabilistic Index*". This index, for a given page image, provides the probability that a word appears in every word-sized bounding box of the image. We show that the textual content information features extracted from these indexes can greatly improve the classification accuracy of any HTS approach. We review how these novel text content features can be extracted for use in specific tasks and show that they do actually boost the performance of our proposed HTS approach.

The calculation of such word probabilistic indexes, was developed as part of approaches to perform word-segmentation-free Key Word Spotting systems [2, 14, 21, 28, 29]. With these indexes we now have probabilistic information of the textual contents of the page, without the need to perform a detailed text base line detection, and performing full Handwritten Text Recognition on the detected text lines.

Although this approach does require some measure of text detection to be performed, it yields adequate results with automatically detected foreground text areas with no human review required. Furthermore, this type of features might not be required for all tasks, but only needed to perform text region classification in corpora that do not present visible graphical clues to allow such a classification. We will see an example of usage of these text content based features in Sec. 7.8. An example of these word probabilistic indexes for a sample image area can be seen in Fig. 3.10. The index contains an entry for each word found, the coordinates of the bounding box and the probability with which it has detected the word.

```
<KWindex pageID="FRCHANJJ_JJ070_0040R_A-chanceryDemo38.jpg">
<spot kw="A"              s=1.000  x=872  y=230  w=53  h=29  gt=1 />
<spot kw="A"              s=1.000  x=844  y=72   w=23  h=47  gt=1 />
<spot kw="AVENIR"         s=1.000  x=1054 y=72   w=100 h=47  gt=1 />
<spot kw="AVOIT"          s=1.000  x=1135 y=181  w=82  h=32  gt=1 />
<spot kw="CE"             s=1.000  x=1303 y=72   w=40  h=47  gt=1 />
<spot kw="COMME"          s=1.000  x=1194 y=72   w=70  h=47  gt=1 />
...
<spot kw="NOSTRE"         s=1.000  x=1172 y=124  w=83  h=37  gt=1 />
<spot kw="NOUS"           s=1.000  x=445  y=181  w=74  h=32  gt=1 />
<spot kw="NOUS"           s=1.000  x=887  y=124  w=79  h=37  gt=1 />
...
<spot kw="QUE"            s=1.000  x=1134 y=72   w=53  h=47  gt=1 />
<spot kw="QUELLE"         s=1.000  x=752  y=124  w=83  h=37  gt=1 />
<spot kw="RECEVEUR"       s=1.000  x=1268 y=124  w=121 h=37  gt=1 />
<spot kw="ROY"            s=1.000  x=468  y=72   w=61  h=47  gt=0 />
...
<spot kw="TOUZ"           s=1.000  x=878  y=72   w=65  h=47  gt=0 />
<spot kw="UNE"            s=1.000  x=739  y=181  w=53  h=32  gt=1 />
<spot kw="VILLE"          s=1.000  x=1289 y=181  w=66  h=32  gt=1 />
<spot kw="FEU"            s=0.999  x=547  y=230  w=35  h=29  gt=1 />
<spot kw="MAISON"         s=0.999  x=820  y=181  w=87  h=32  gt=1 />
<spot kw="POIER"          s=0.997  x=512  y=181  w=70  h=32  gt=1 />
<spot kw="TENUZ"          s=0.997  x=238  y=124  w=74  h=37  gt=1 />
<spot kw="CRIER"          s=0.996  x=195  y=181  w=70  h=32  gt=1 />
<spot kw="TENANT"         s=0.996  x=235  y=230  w=87  h=29  gt=1 />
<spot kw="PROQUIN"        s=0.982  x=624  y=230  w=138 h=29  gt=0 />
<spot kw="LODUN"          s=0.980  x=1388 y=181  w=74  h=32  gt=1 />
<spot kw="PART"           s=0.972  x=356  y=230  w=48  h=29  gt=1 />
<spot kw="D'ARGENT"       s=0.958  x=597  y=124  w=108 h=37  gt=1 />
<spot kw="ESTIENNE"       s=0.955  x=1047 y=181  w=104 h=32  gt=1 />
<spot kw="AIT"            s=0.953  x=1446 y=124  w=83  h=37  gt=1 />
<spot kw="VENTE"          s=0.914  x=300  y=181  w=74  h=32  gt=1 />
<spot kw="TROIZ"          s=0.902  x=147  y=230  w=91  h=29  gt=0 />
<spot kw="DITE"           s=0.895  x=632  y=181  w=44  h=32  gt=1 />
<spot kw="PHILIPPES"      s=0.885  x=103  y=72   w=113 h=47  gt=1 />
...
<spot kw="METTRE"         s=0.798  x=98   y=181  w=83  h=32  gt=1 />
<spot kw="PART"           s=0.667  x=990  y=72   w=40  h=47  gt=0 />
<spot kw="CERTEINE"       s=0.625  x=387  y=124  w=168 h=37  gt=0 />
<spot kw="BOURS"          s=0.614  x=1399 y=124  w=83  h=37  gt=0 />
<spot kw="QUE"            s=0.611  x=332  y=124  w=57  h=37  gt=0 />
<spot kw="SATISFAIRE"     s=0.584  x=1075 y=124  w=113 h=37  gt=1 />
<spot kw="ESCUNE"         s=0.450  x=1397 y=72   w=74  h=47  gt=0 />
...
<spot kw="FAT"            s=0.100  x=1523 y=124  w=185 h=37  gt=0 />
<spot kw="CROIZ"          s=0.087  x=147  y=230  w=91  h=29  gt=1 />
<spot kw="JOURONDE"       s=0.077  x=1516 y=181  w=203 h=32  gt=0 />
...
<spot kw="MAVANT"         s=0.001  x=1296 y=230  w=104 h=29  gt=0 />
<spot kw="PHILIPS"        s=0.001  x=105  y=72   w=117 h=47  gt=0 />
<spot kw="SAATISFAIRE"    s=0.001  x=1075 y=124  w=113 h=37  gt=0 />
<!- 536 spots, 94 rw; Density:  536/94 = 5.7 spots per running
word.->
</KWindex>
```



Philippes par la grace de Dieu roys de France. Savoir faisons à tous presens et avenir que comme pour ce que Estienne Hervé nous estoit tenuz en une certeinne somme d'argent de la quelle il ne nous povoit satisfaire nostre receveur de Tours ait fait mettre et crier en vente pour nous poier de la dite somme une maison que le dit Estienne avoit en la ville de Lodun ou bourc Sainte Croiz tenant d'une part à la maison feu Perrequin et d'autre part à la maison Estienne le Huger mouvant du chapitre de

Figure 3.10: Indexed text image region and its ground-truth transcript (from the Chancery corpus Sec. 5.10).

Unfortunately these word posteriors can not just be directly included to the feature vector. The sheer size of the vocabulary makes this an impractical idea. Furthermore, in order to have a positive impact on the performance of our HMM models we must present this data in an adequate manner. Namely, as additional vector components to accompany the graphical features.

Therefore an adequate word vocabulary needs to be selected that allows the models to better differentiate between the chosen *LEs* for the task at hand. In a simple region detection case a simple count of the indexed words for each pixel line could provide information regarding the presence of text, but this would be an excessive and improper use of these features. To tackle complicated cases that require region classification we need vocabularies that capture the usual words and sentence structure used in each type of *LEs*. For example, we could review the words used at the beginning, middle and ending of a specific type of record, in order to have a vocabulary list that allows us to differentiate between them. This specific vocabulary can be calculated in an automatic manner by means of Linear Discriminant Analysis (LDA) [8] or with the help of an expert.

Once the vocabulary that allows us to distinguish between *LEs* is defined, we can calculate feature values that can help us determine to which *LE* does the current pixel row belong to. For each pixel line we review which detected index entries bounding boxes contain it, we look at the word in each entry and if it pertains to the vocabulary subset of any of the *LE* (and surpasses a minimum probability value) we add the probability to the adequate histogram count.

The idea is that the histogram count feature value for a specific *LE* will be larger than the count for the rest of *LEs* in the page area where that element is present. Continuing with the example described for Fig. 3.9 we determined that in order to differentiate when a record starts or ends we have three specific *LEs*: mid record (M), start of the record (I) and ending of the record (F). Since the graphical features will not suffice to differentiate between these *LEs*, we defined a vocabulary that aids us to do so. Using the probabilistic indexes seen in Fig. 3.10 we computed for each pixel line the histogram count for each *LE* and use them as feature values. In Fig. 3.11 we can see the resulting feature vectors for the three *LEs* drawn next to the page used to calculate them with some of the probabilistic word index entries that influenced the final value highlighted with boxes. The use of these text content based features can greatly improve the performance of text region classification as we will see later in Sec. 7.8.

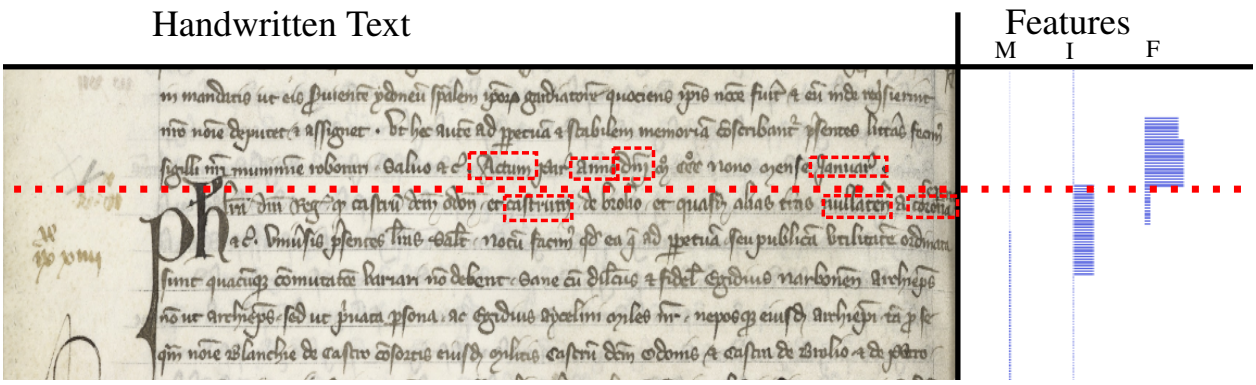


Figure 3.11: Illustration of the resulting histogram count for three layout elements side-by-side with the image portion it was calculated on. The horizontal dotted line represents the actual division between the two acts. We can see how the histogram count for the end of the record words (F) increases just before the end of the record (delimited by the red dotted line) and how the histogram count for starting of act words (I) increases at the beginning of the next record. In the handwritten text we highlight with boxes some key entries that have caused this change in the feature values.

3.7 Chapter Conclusions

In this chapter I have presented my text region detection and classification approach. I have presented the classical image preprocessing techniques used and also the classical region detection approach our method is compared against.I have formally presented the modelling considered to apply HMM to STRDC.

This chapter also includes the strategy to effectively use our probabilistic method in real production scenarios. This strategy allows us to produce ground-truth like quality regions while minimizing the amount of manually-labelled training data and the required user review time.

Additionally, I have laid out our adaptation on the usage of Convolutional Neural Networks and Hidden Markov Models in a simple tandem model for HTS tasks. Finally, I have delineated the novel work on text content based features that can be calculated from an automatically obtained probabilistic word index. These text content based features will allow us to tackle tasks where graphical information is not enough.

Bibliography

[1] Adiguzel, H., Sahin, E., and Duygulu, P. (2012). A hybrid for line segmentation in handwritten documents. In *Proocedings of ICFHR*, pages 503–508.

[2] Bluche, T., Hamel, S., Kermorvant, C., Puigcerver, J., Stutzmann, D., Toselli, A. H., and Vidal, E. (2017). Preparatory kws experiments for large-scale indexing of a vast medieval manuscript

collection in the himanis project. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 311–316.

[3] Bluche, T., Ney, H., and Kermorvant, C. (2013). Tandem hmm with convolutional neural network for handwritten word recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2390–2394.

[4] Breuel, T. M. (2002). Two geometric algorithms for layout analysis. In Lopresti, D., Hu, J., and Kashi, R., editors, *Document Analysis Systems V*, pages 188–199, Berlin, Heidelberg. Springer Berlin Heidelberg.

[5] Dechter, R. (1986). Learning while searching in constraint-satisfaction-problems. pages 178–185.

[6] Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2017). A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, 64:1 – 14.

[7] Fernández, D., Lladós, J., Fornés, A., and Manmatha, R. (2012). On influence of line segmentation in efficient word segmentation in old manuscripts. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 763–768.

[8] Haeb-Umbach, R. and Ney, H. (1992). Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 13–16 vol.1.

[9] Hermansky, H., Ellis, D. P. W., and Sharma, S. (2000). Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 3, pages 1635–1638 vol.3.

[10] i Gadea, M. P., Toselli, A. H., and Vidal, E. (2004). Projection profile based algorithm for slant removal. In *Proceedings of ICIAR*.

[11] Jelinek, F. (1998). *Statistical methods for speech recognition*. MIT Press.

[12] Kavallieratou, E. and Stamatatos, E. (2006). Improving the quality of degraded document images. In *Document Image Analysis for Libraries, 2006. DIAL '06. Second International Conference on*, pages 10 pp. –349.

[13] Kesiman, M. W. A., Burie, J. C., and Ogier, J. M. (2016). A new scheme for text line and character segmentation from gray scale images of palm leaf manuscript. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 325–330.

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

[15] Liang, J., Ha, J., Haralick, R. M., and Phillips, I. T. (1996). Document layout structure extraction using bounding boxes of different entitles. In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pages 278–283.

[16] Likforman-Sulem, L., Zahour, A., and Taconet, B. (2007). Text line segmentation of historical documents: a survey. *Int. J. Doc. Anal. Recognit.*, 9:123–138.

[17] Manmatha, R. and Srimal, N. (1999). Scale space technique for word segmentation in handwritten documents. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, SCALE-SPACE '99, pages 22–33, London, UK. Springer-Verlag.

[18] Messaoud, I., Amiri, H., Abed, H., and Margner, V. (2012). A multilevel text-line segmentation framework for handwritten historical documents. In *Proocedings of ICFHR*, pages 515–520.

[19] Moysset, B., Bluche, T., Knibbe, M., Mohamed Faouzi Benzeghiba, R. M., Louradour, J., and Kermorvant, C. (2014). The A2iA multi-lingual text recognition system at the second maurdor evaluation. In *Proocedings of ICFHR*, pages 297–302.

[20] Oliveira, S. A., Seguin, B., and Kaplan, F. (2018). dhsegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 7–12.

[21] Puigcerver, J. (2018). *A Probabilistic Formulation of Keyword Spotting*. PhD thesis, Universitat Politècnica de València.

[22] Quirós, L. (2018). Multi-task handwritten document layout analysis. *CoRR*, abs/1806.08852.

[23] Romero, V., Fornés, A., Serrano, N., Sánchez, J. A., Toselli, A. H., Frinken, V., Vidal, E., and Lladós, J. (2013). The esposalles database: An ancient marriage license corpus for off-line handwriting recognition. *Pattern Recognition*, 46(6):1658 – 1669.

[24] Smith, R. W. (2009). Hybrid page layout analysis via tab-stop detection. In *2009 10th International Conference on Document Analysis and Recognition*, pages 241–245.

[25] Stahlberg, F. and Vogel, S. (2015). Detecting dense foreground stripes in arabic handwriting for accurate baseline positioning. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 361–365.

[26] Surinta, O., Holtkamp, M., Karabaa, F., van Oosten, J.-P., Schomaker, L., and Wiering, M. (2014). A* path planning for line segmentation of handwritten documents. In *Proocedings of ICFHR*, pages 175–180.

[27] Toselli, A. H., Romero, V., and Vidal, E. (2011). *Language Technology for Cultural Heritage*, chapter Alignment between Text Images and their Transcripts for Handwritten Documents., pages 23–37. Theory and Applications of Natural Language Processing. Springer,. Caroline Sporleder, Antal van den Bosch y Kalliopi Zervanou (Eds.).

[28] Toselli, A. H., Vidal, E., Puigcerver, J., and Noya-García, E. (2018). Probabilistic multi-word spotting in handwritten text images. *Pattern Analysis and Applications*.

[29] Toselli, A. H., Vidal, E., Romero, V., and Frinken, V. (2016). HMM word graph based keyword spotting in handwritten document images. *Information Sciences*, 370-371:497 – 518.

[30] Vidal, E., Thollard, F., De La Higuera, C., Casacuberta, F., and Carrasco, R. C. (2005). Probabilistic finite-state machines – parts i & ii. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(7):1013–1039.

[31] Villegas, M. and Toselli, A. H. (2014). Bleed-through Removal by Learning a Discriminative Color Channel. In *Frontiers in Handwriting Recognition (ICFHR), 2014 International Conference on*, pages 47–52.

[32] Wong, K. Y. and Wahl, F. M. (1982). Document analysis system. *IBM Journal of Research and Development*, 26:647–656.

[33] Young, S., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1997). *The HTK Book: Hidden Markov Models Toolkit V2.1*. Cambridge Research Laboratory Ltd.

CHAPTER 4

TEXT REGION EXTRACTION

Chapter Outline

4.1 Introduction

As we indicated in chapter 3 we envision HTS as a two step process. Firstly, the text regions are to be detected (and possibly classified) and secondly, the yielded detection result is used in order to perform the extraction.

Although this two step separation might now seem obvious, classically this was not the case and most approaches used to mix both steps (see Chap. 2). This shift is specially visible if we observe the recent evolution of the handwriting segmentation competitions.

The aforementioned shift in the community leads us to define our text region extraction algorithm so that it makes the best use of the information yielded by the detection system. We developed a text region extraction technique that aims to calculate an equidistant extraction frontier as per the detection boundaries of two adjacent regions.

We developed a robust binarization-free approach inspired in path planning algorithms that uses the detected boundaries to restrict the search areas. It is important to note that although our STRDC system yields straight detection boundaries the method presented for the extraction approach is not limited to these type of boundaries and can make use of piecewise linear boundaries.

This method can be used both in handwritten documents as well as in printed texts. We use the concept of distance maps [4] in order to obtain results that are as separate for each text line body as possible with the aim to provide an equidistant frontier when possible. Equidistant separation frontiers provide adequate properties when tackling complex scripts with large amounts of disjoint strokes and diacritical marks.

As the most typical use of this technique is text line extraction (a particular case of the text region extraction problem) we have adapted the vocabulary to this specific task. Hence, we will mainly talk about text line regions in this chapter, although the technique can be applied to extract any type of region. In fact this method was initially developed for text line extraction but it can be used to actually calculate a frontier of any type of text regions given a detection boundary.

4.2 Reference Text Extraction Method

As we did for the text region detection subtask it is important that we define a reference approach for text line extraction. Although the most important comparisons to perform are against other existing methods we have a very necessary use for our own basic extraction method.

The basic method performs a straightforward projection from the already existing baselines in order to compute the extraction polygon, which just computes its area by taking some pixels above and below along the path of the baseline [3]. In Fig. 4.1 we can observe some sample extraction polygon calculations computed with this reference method for two types of baselines.

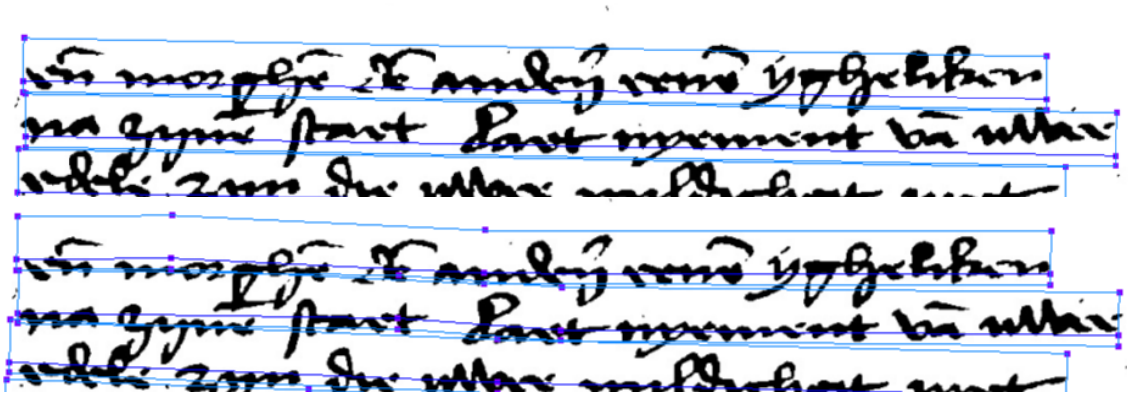


Figure 4.1: Automatic extraction polygon generated automatically from straight baselines (above) and piece wise curve baselines (below).

This simple projection method will allow us to measure how important is the actual computation of the text line extraction polygon in comparison to the text baseline detection task. Since this technique does not perform any sort of intelligent decisions, it allows us to evaluate how much of the final segmentation accuracy is due to the results of the detection approach and what is the added value provided by the extraction technique.

4.3 Distance Maps

As said at the start of this section our method for text extraction aims to achieve equidistant extraction frontiers when possible. This objective is reached by the idea of a distance model. The definition of the distance model used in our approach is inspired by the *DTOCS* and *WDTOCS* ideas [4].

Given a generic grey-scale image, we perform a distance calculation with real numbers and generate a distance map, in which the value of each pixel position is the length of the shortest path to the nearest *foreground pixel*. This discrete 8-path takes into consideration the grey value of the pixels along the path.

Let $X \subset \mathbb{Z}^2$ be a 2-dimensional discrete space. Let $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in X$ be two points of this discrete space. Let $\Psi_X(\boldsymbol{x}, \boldsymbol{y})$ be the set of 8-connected paths in X linking $\boldsymbol{x}$ and $\boldsymbol{y}$. Let $\gamma = \boldsymbol{a}_1, ..., \boldsymbol{a}_i, \boldsymbol{a}_{i+1}, ..., \boldsymbol{a}_n \in \Psi_X(x, y)$ be a path composed of n pixel coordinates. Let $\mathcal{G}_X(\boldsymbol{a}_i) \in [0, 1]$ denote the grey value of pixel $\boldsymbol{a}_i$.

We define the distance between $\boldsymbol{a}_i$ and $\boldsymbol{a}_{i+1}$ as $d_X(\boldsymbol{a}_i, \boldsymbol{a}_{i+1}) = \delta_X(\boldsymbol{a}_i, \boldsymbol{a}_{i+1}) + \mathcal{G}_X(\boldsymbol{a}_{i+1})$ where $\delta_X(\boldsymbol{a}_i, \boldsymbol{a}_{i+1})$ is the Euclidean distance between them[a].

[a]Given that the calculation will be performed between adjacent pixels in a discrete 8-path connected matrix the values will be either 1 or $\sqrt{2}$

Thus, we can define the length of the path γ as $\Lambda(\gamma) = \sum_{i=1}^{n-1} d_X(\boldsymbol{a}_i, \boldsymbol{a}_{i+1})$. With this, we can define the distance between the two points $\boldsymbol{x}$ and $\boldsymbol{y}$ as $D_X(\boldsymbol{x}, \boldsymbol{y}) = \min_{\gamma \in \Psi_X(\boldsymbol{x}, \boldsymbol{y})} (\Lambda(\gamma))$. Leaving the final distance map definition as:

$$\mathcal{F}_X(\boldsymbol{x}) = \begin{cases} 0 & \boldsymbol{x} \in \theta(X) \\ \min_{y \in \theta X} D_X(\boldsymbol{x}, \boldsymbol{y}) & \boldsymbol{x} \notin \theta(X) \end{cases} \tag{4.1}$$

where $\theta(X)$ are the set of foreground pixels that can be obtained with any binarization method of the input image X. Please note that although we use the foreground pixels as the subset of pixels in the image to which we must calculate the distance, the actual definition of the path distance does include all grey values present in the image. This distance map can be easily computed by a sequential two-pass algorithm as described in [4]. A result of the distance map calculation can be seen in Fig. 4.2.

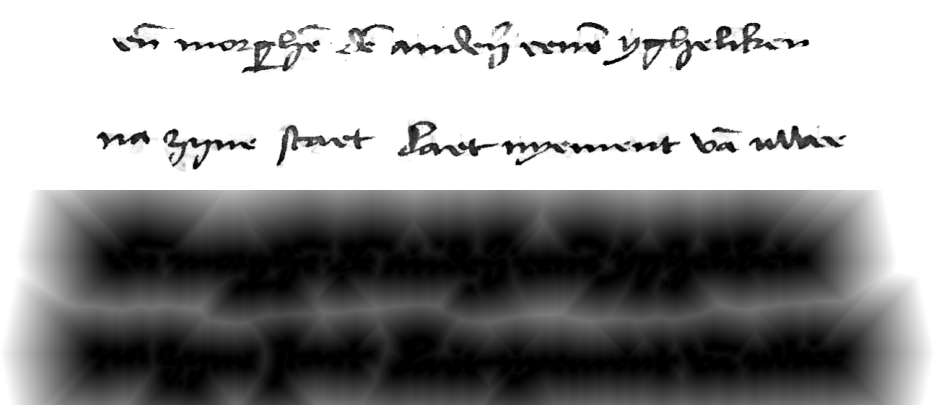


Figure 4.2: Visualization of the distance map (normalized to standard intensity map range) result for a part of an image. In this sample image the equidistant frontier can be easily seen as the valley of low values between the two bodies of the text lines.

4.4 Baselines Usage

Given the generated *Distance Map* $\mathcal{F}_X$ of an image, we proceed to calculate the separation frontiers. Since a text image contains N text lines (or regions) we need to calculate the $N+1$ separation frontiers between each adjacent text line. As there are many separate paths that can transverse the distance map, we must calculate the shortest one contained between each pair of adjacent lines $\boldsymbol{l}_i, \boldsymbol{l}_{i+1} : 1 \leq i < N$. We launch our dynamic programming path finder in a restricted area comprehended by these text lines: $\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1}) \subseteq X$.

This restricted area can be easily defined with the list of detected baselines taken as input by our approach. As commented earlier, the set of baselines can be automatically detected by any text line detection algorithm, manually annotated or a mixture of both. We use the points in the baselines to calculate a rectangle region containing them all [2]. Furthermore, the baselines are also used as geometric restrictions that cannot be crossed.

4.5 Calculation of the Extraction Polygon

Given a search area $\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1})$, the possible frontiers between two text lines are the set of digital 8-connected paths: $\Gamma(\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1}))$. Where each of these paths in this set traverse the search area from start to finish.

If we consider a specific path $\theta \in \Gamma(\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1}))$, we define its total distance as the sum of the distances between each adjacent points contained in the path. Given two adjacent points $\boldsymbol{a}_j \in \theta$ and $\boldsymbol{a}_{j+1} \in \theta$, we define the distance between them as:

$$p_X(\boldsymbol{a}_j, \boldsymbol{a}_{j+1}) = \delta_X(\boldsymbol{a}_j, \boldsymbol{a}_{j+1}) + \frac{\max(\mathcal{F}_X[\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1})]) - \mathcal{F}_X(\boldsymbol{a}_{j+1})}{\max(\mathcal{F}_X[\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1})])}. \tag{4.2}$$

Our separation frontier will be the pathway existing in $\Gamma(\mathcal{A}_X(\boldsymbol{l}_i, \boldsymbol{l}_{i+1}))$ that minimizes the above defined distance. This 8-connected path way will be the shortest path to transverse the search area while aiming to be as separate to the bodies of both text lines as possible. This is true by definition since the defined distance contains both a term for euclidean distance and a term that considers the distance to the foreground and grey area pixels that conform the text line bodies.

In order to calculate the possible set of paths Γ we see the search area as an 8-connected graph where each of the pixel positions is a node of the graph. As the paths must transverse the search area, we perform such a calculation from left to right, but it could be done in any orientation. Starting from the left border pixels of the search area we update the following column of pixels as per the left side productions of the 8-connected graph. This calculation is performed by applying the $p_X(\boldsymbol{a}_i, \boldsymbol{a}_{i+1})$ function to each cell; imitating the classic dynamic programming flood fill algorithm strategy.

This of course has issues as sometimes the minimal cost route requires to be able to backtrack or go up-down along a column (Fig. 4.3). In order to resolve this, once all nodes have been visited through the left-hand side productions, a review process is launched from right to left. This is performed in order to re-estimate the optimal production to reach each node considering the right-hand side and top-down productions. These *forwards-backwards* calculations are performed iteratively (usually 2-3 passes in average) until no changes occur on the optimal path to each of the nodes is calculated.

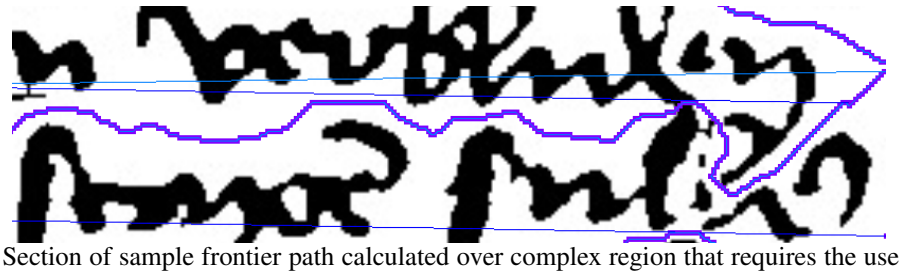


Figure 4.3: Section of sample frontier path calculated over complex region that requires the use of top-down and right productions

Once the algorithm converges, in order to retrieve the minimum cost path we just have to select the node on the last right hand side column of the search area that was reached with the least

cost p_X. From this node, we backtrack through the solution matrix until reaching one of the left hand border start nodes. As each of the frontier calculations are independent from each other this algorithm can be trivially parallelized.

4.6 Extraction Frontier Collision Resolution

In historical handwritten texts it is very usual that the ascenders and descenders of adjacent lines touch. This leaves no possibility to calculate an extraction polygon that does not cross through any of the pixels that compose the text line (Fig. 4.4). These issues force some methods to include special ways to resolve them [1].

Our approach avoids this type of issues completely since our path finding approach does not use foreground pixels as hard borders. Our method simply avoids the body of the text line and neighbouring area as much as possible due to the cost associated in crossing such nodes. Hence, if all possible paths that transverse the search area must cross through a foreground pixel, the algorithm will simply select the path that crosses the minimum amount of foreground pixels while being as distant as possible to the text lines bodies.

Figure 4.4: Section of sample frontier path that is forcefully required to cross through foreground pixels and the re-estimated circular frontier correction

Furthermore, once the optimal frontier path is calculated it can be reviewed for foreground collision points. If the optimal path is found to have collision points we have the opportunity to detect such cases and adjust the output result to resolve the issue.

In our case, we performed an easy resolution by adjusting the frontier by means of a circular mask. The mask is used to assign the context in the circular area to both the affected upper and lower extraction polygons. This approach of assigning the conflicting area to both text lines is usually far more valid than trying to perform a detailed issue resolution with no extra information. If we consider TLE as part of a preprocessing pipeline required for HTR and KWS tasks, the systems used are much more robust to the noise of incorrect artefacts that made it to the final text line image than to a deliberate reduction of information/context.

4.7 Chapter Conclusions

In this chapter I have presented a robust binarization-free approach inspired in path planning algorithms that uses the detected boundaries yielded by the method described in Chapter 3 to restrict the search areas.

The method aims via the use of distance maps to calculate an extraction frontier that is as separate of each region body as possible with the aim to provide an actual equidistant frontier when feasible. The method shown provides a solution even when the foreground elements of each region touch and allows for additional correction methods to be applied in such cases.

Bibliography

[1] Fernández-Mota, D., Lladós, J., and Fornés, A. (2014). A graph-based approach for segmenting touching lines in historical handwritten documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):293–312.

[2] Freeman, H. and Shapira, R. (1975). Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM*, 18(7):409–413.

[3] Romero, V., Sánchez, J. A., Bosch, V., Depuydt, K., and de Does, J. (2015). Influence of text line segmentation in handwritten text recognition. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 536–540.

[4] Toivanen, P. J. (1996). New geodosic distance transforms for gray-scale images. *Pattern Recognition Letters*, 17(5):437–450.

CHAPTER 5

CORPORA

Chapter Outline

5.1 Introduction

In this chapter we present the different corpora used in the different experimentation performed during the course of this thesis.

In order to validate or correctly assess the proposed methods described in Chapters 3 and 4 we required a set of corpora. Each of the used corpus presents a different scenario in which to evaluate different aspects of the techniques covered in this doctoral dissertation.

5.2 Motivation

This thesis chapter contains a description of the different corpora used throughout the experimentation performed during the research. It is important that the reader notes that due to the choices performed during the described research, as indicated in Chapter 3, a need for corpus with specific characteristics is warranted:

- Due to the fact that our primary research focus is region type classification we require corpora that have labelled lists of the contents of each page.
- As we also provide (as a sub-product) the coordinates for the baseline we require corpora that have baselines for the text lines or segmentation lines for page regions.
- We also require, for some experiments, to have access to the actual text of each of the text lines. We require this in order to measure how our approach impact higher level tasks as we describe in Sec. 6.4.

The fact that most standard corpora for region or text line segmentation are not compliant with the above listed requirements forced us to procure our own. At the time the research for this thesis was started, the community were not actively tackling region detection as an independent task. Region and text line classification were not even considered.

The document layout analysis community was originally more concerned with text line segmentation as a whole. This can be observed in the competition that dominated the area for seven years [2–4, 9]. Its evaluation measure and the competition corpus was used widely during a period of time. The community has shifted, since then, its focus more on region and text line detection as can be seen in the more modern competitions [1, 5].

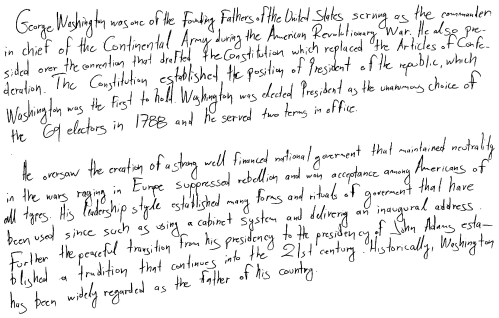
For each different corpus used during our research a different Layout-Region dictionary was used as per the task that was being resolved for that particular corpus. Next we will detail each of the corpus used during this doctoral dissertation.

5.3 Handwriting Segmentation Contest Corpus- 2013 edition

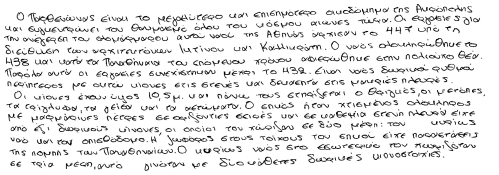
The **Handwriting Segmentation Contest** and its associated corpus and evaluation first appeared in 2007 [2]. After this first appearance it has been re-edited a few times [3, 4, 9]. It can be considered the first successful attempt to unify the community by providing a corpus and evaluation measures in order to compare different text line segmentation methods.

Throughout its different editions improvements and changes have been performed to the evaluation measure Sec. 6.8 and the contents of the corpus. In our case we have chosen to use the latest corpus set from 2013 [9].

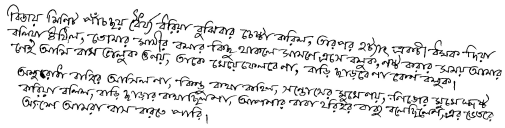
The *Handwriting Segmentation Contest 2013 Corpus* is a synthetic corpus created for competition purposes. Both the training and test section of this corpus consists of a collection of 150 independently written pages by different individuals. The corpus also presents the added difficulty of being multi-script as one third of the total number of pages, is written in English, Greek or Bangla. We denote it as a synthetic corpus because the text written in each page was the same sample text provided to the students. Furthermore, their lack of intent to actually communicate through their writing is visible in the sloppy penmanship. Sample of handwritten document images of this dataset can be seen in Figure 5.1.



(a) English page sample



(b) Greek page sample



(c) Bangla page sample

Figure 5.1: Sample page images of the different scripts found in the ICDAR 2013 edition of the Handwriting Segmentation Contest Corpus.

From the layout perspective this corpus can be considered to be labelled at pixel level. The groundtruth provided for this contest were raw data image files. The files contained a matrix of integer values: with zeros corresponding to pixel positions for background and positive integer values to tag pixels that were part of one of the foreground regions.

Although this is an interesting corpus for measuring *Text Line Extraction* algorithms we do not consider it adequate to evaluate accuracy of *Text Line Detection*. The artificially added complexity due to its synthetic nature makes it unrealistic. Additionally we will later discuss the issues found out with the proposed evaluation measure in Chp. 7.

As we discussed in Chapter 3 we consider *Text Line Segmentation* as composed by a *Text Line Detection* and a *Text Line Extraction* subtasks. Where the extraction depends on the results yielded by the detection. The detailed level of labelling, the three different types of scripts used and its multi writer feature makes it an adequate and challenging corpus to measure *Text Line Extraction* algorithms.

As part of our research work we have manually created the baseline groundtruth for this corpus. The baselines were done in a very simplistic manner trying to emulate a possible output of a baseline detection system. Up to four control points were used to describe the text base lines. A sample of a handwritten document image of these dataset with the created baselines can be seen in Figure 5.2.

This groundtruth was created in order to allow researchers to measure the performance of a text line extraction approach in an independent manner by isolating it from the errors that could be introduced from automatically detected baselines.

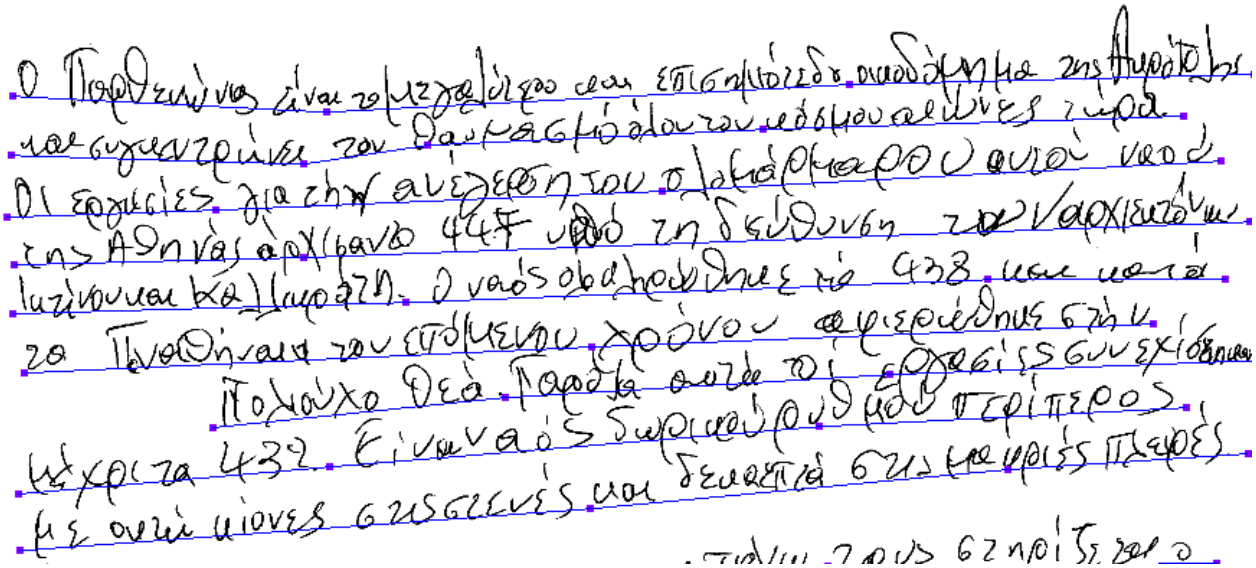


Figure 5.2: Sample page region with the simple baseline groundtruth created for it.

5.4 Hattem

The *C5 Hattem Manuscript* is a single writer document from the 15th century [8]. This document was tackled as part of the *tranScriptorium* project and was part of the Dutch language data to be processed. The document is mainly written in Middle Dutch and composed by 572 sheets[a].

Figure 5.3: Page image examples of the Hattem collection.

In our research we used a set of 40 pages from the complete collection. This subset of pages was initially selected and used used in [8]. The selected pages are not consecutive in the book and they were selected from the complete collection by an expert. This selection was performed in order to obtain a reduced set of pages that would be representative of all the different page formats that appear along all the book. Figure 5.3 shows some examples of the selected images.

The 40 pages were manually transcribed [b] by an expert palaeographer. As explained in [8], this medieval manuscript contains a lot of abbreviations, which are indicated by special symbols no longer in use.

[a] Utrecht university library, MV: C5, http://objects.library.uu.nl/reader/index.php?obj=1874-44915&lan=en
[b] http://wemal.let.uu.nl/hattem-c5.html

Table 5.1: Basic statistics of the different partitions in the selected 40 pages of the Hattem dataset.

Number of:	P0	P1	P2	P3	P4	P5	P6	P7	Total
Pages	5	5	5	5	5	5	5	5	40
Lines	186	200	195	185	191	191	203	201	1,552
Run. words	1,280	1,351	1,318	1,227	1,237	1,230	1,344	1,343	10,330
Run. OOV	261	289	275	275	265	294	332	268	-
Lex. OOV	235	255	241	247	224	231	290	241	-
Lexicon	567	594	584	599	576	563	639	586	2,751
Character set size	45	43	44	46	43	52	44	45	60

From the layout point of view, the selected document images have a single column. The lines are very close each other and there are many occurrences of touching ascenders and descenders in consecutive lines. The line localization and the line extraction seem very difficult because of this reason.

The groundtruth transcripts are annotated with the abbreviation symbol used and also with the hypothesized expansion of the abbreviation [6].

The 40 pages were divided into 8 blocks of 5 pages each, aimed at performing cross-validation experiments. Table 5.1 contains some basic statistics of the different partitions defined. The number of running words for each partition that did not appear in the other seven partitions is shown in the running out-of-vocabulary (Run. OOV) row. The coverage percentage of running OOV is 22%.

From the layout perspective the groundtruth was performed at various levels. The text lines present in the text were annotated in three different levels of detail: *Polygon*, *Poly-baseline* and *Straight-baseline*.

In the *Polygon* method, a polygon surrounding perfectly each text line was obtained. A detection process was automatically carried out in the 40 selected Hattem pages and afterwards the system errors were manually corrected. In Fig. 5.4 we can see a sample segment of the final result of the groundtruth at this level.

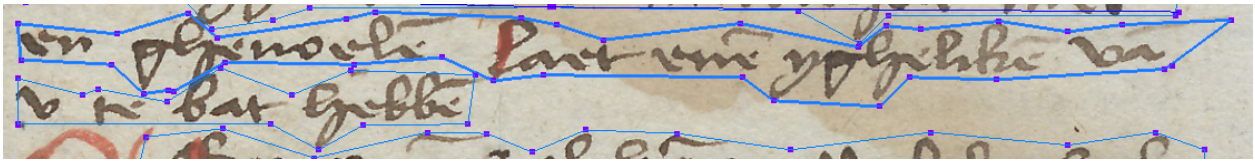


Figure 5.4: Sample image of a Hattem collection page segment with the surrounding groundtruth polygon.

In the *Poly-baseline* approach, a poly-line annotating the baseline of each text line was automatically generated and then manually corrected. See Fig. 5.5 for a sample segment of a page presenting this type of annotation.

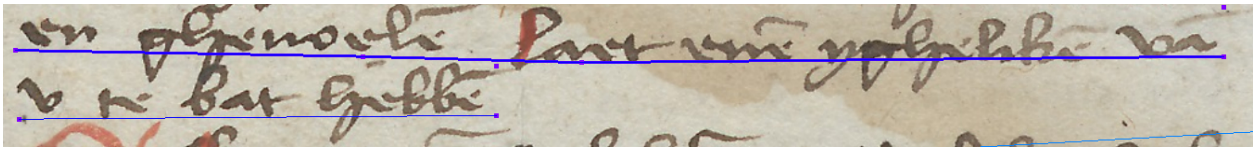


Figure 5.5: Sample image of a Hattem collection page segment with the poly-baseline groundtruth annotation.

In the last method studied, a horizontal *Straight-baseline* was obtained for each localized text line. Fig. 5.6 presents a sample page segment with this simple baseline annotation.

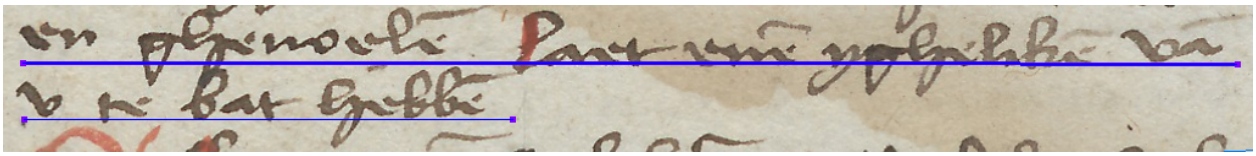


Figure 5.6: Sample image of a Hattem collection page segment with the straight baseline groundtruth annotation.

It is also important to note that the *Polygon* approach can be considered to all effects the actual text line segmentation groundtruth for this corpus as it was manually reviewed and allows anyone to extract the text line images without error.

5.5 Cristo Salvador

Cristo Salvador corpus (or CHRIS for short) was compiled from a XIX century small book with 52 colour images of text pages, written by a single writer and scanned at 300 dpi. The manuscript was kindly provided by the *Biblioteca Valenciana Digital* (BiVaLDi)[c]. Examples pages from this corpus can be seen in Fig. 5.7.

The book was written by Vicente Boix chronicler of Valencia city. In it, Vicente, describes the different festivities that took place in Valencia due to the six century anniversary of the *Salvador* image arriving to the city.

[c] http://bivaldi.gva.es/es/consulta/registro.cmd?id=281.

Figure 5.7: Examples of page images from CS corpus.

We adopted the predefined *"book"* partition [7], where the training set contains the first 32 page images, and the test set is composed of the 20 remaining pages. During our research, the training partition was further divided into a training set proper and a development set. Table 5.2 shows relevant features of these partitions, along with the distribution *Layout-Regions* defined for this corpus.

Each training page was annotated with a sequence of reference *Layout-Region* labels. For this corpus the following types of *Layout-Elements* were selected:

- Line Body (`LB`): Region occupied by the main body of a normal handwritten text line that runs along the entire width of the text block it belongs to.
- Start Paragraph Line (`PL`): Main body region of a generally indented line, often opening a paragraph.
- Short Line (`SL`): Main body region of a line which is shorter than the normal width of the considered text block. Centred (shorter) lines are typically built on this kind of element and paragraph-closing lines often correspond to this type of line region.
- Inter Line (`IL`): Region spanning between two consecutive text lines, generally crossed by the ascenders and descenders of the adjacent text lines.
- Blank Space (`BS`): Large rectangular region of blank space usually found as top/bottom page margins.

- Non-text region (NT): Stands for everything which does not belong to any of the other regions.

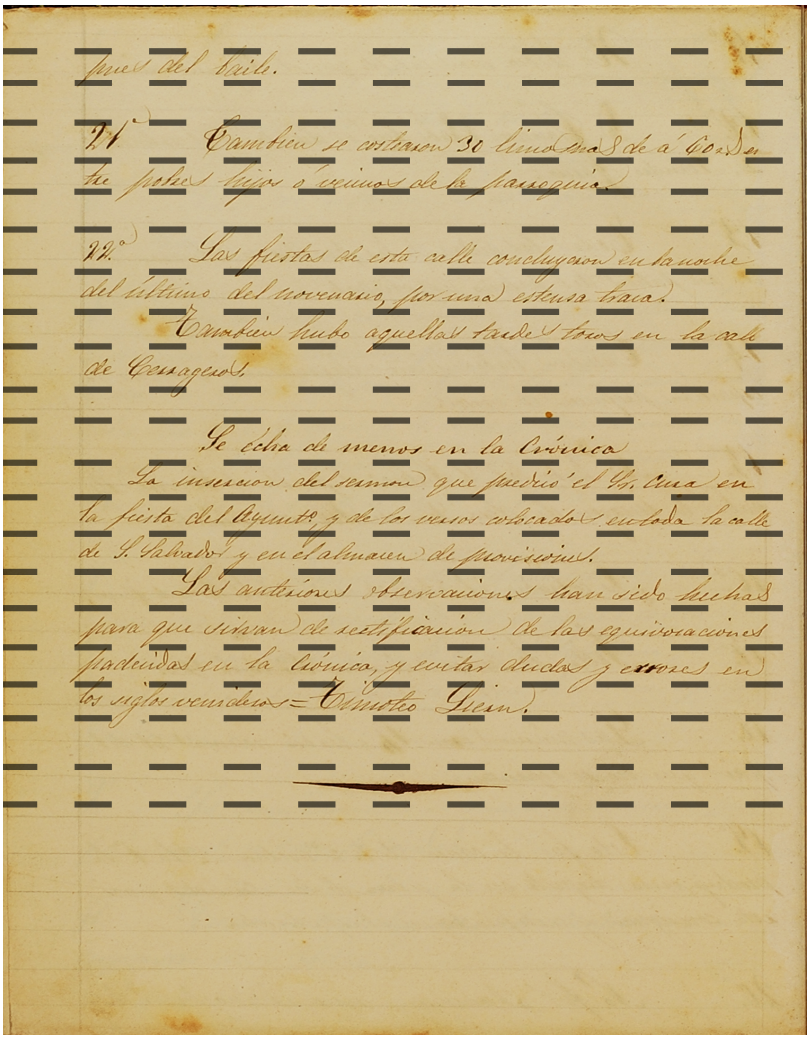


Figure 5.8: Page sample of the *Cristo Salvador* corpus. The sample contains graphical annotations with the groundtruth generated for this corpus: the layout-regions and baselines corresponding to each of the different act region types and transitions

In addition, to create groundtruth for evaluation purposes, vertical line positions of validation and test pages were carefully determined as follows: first, baseline positions were automatically detected then the resulting positions were manually verified, adjusted and/or rectified by a human operator to ensure correctness.

Table 5.2: LR dictionary and statistics of the CHRIS corpus partition used in this work.

Number of:	Train	Devel	Test	Total
Pages	18	14	20	52
Total text line regions	348	337	497	1 182
REGULAR LINES (LB+IL)	325	313	442	1 080
SHORT LINES (SL+IL)	11	12	35	58
PARAGRAPH LINES (PL+IL)	12	12	20	44
BLANK SPACES (BS)	39	34	70	143
NON-TEXT LINES (NT+IL)	9	7	8	24

5.6 Llibres d'Esposalles

Marriage License Books collection, called Llibres d'Esposalles, or WED for short, is conserved at the Archives of the Cathedral of Barcelona.

The full *Esposalles* data set is composed of 291 books with information of approximately 600,000 handwritten marriage licenses given in 250 parishes between years 1451 and 1905. In our experiments, we used only volume 69, which encompasses 173 pages digitized at 300 dpi in true colour [23]. It was divided into training, development and test blocks as shown in Table 5.3. Some examples of text from this corpus can be seen in Fig. 5.9. This corpus was divided into the following partitions:

- Train: pages 1 to 100 used for the training of the HMMs and LMs.
- Development: from page 101 to 150 used during experimentation as validation to review the impact of the training and decoding parameters.
- Test: pages from 151 to 173 are separated from the training and empirical tuning used to perform the final test to assess the performance of our approach.

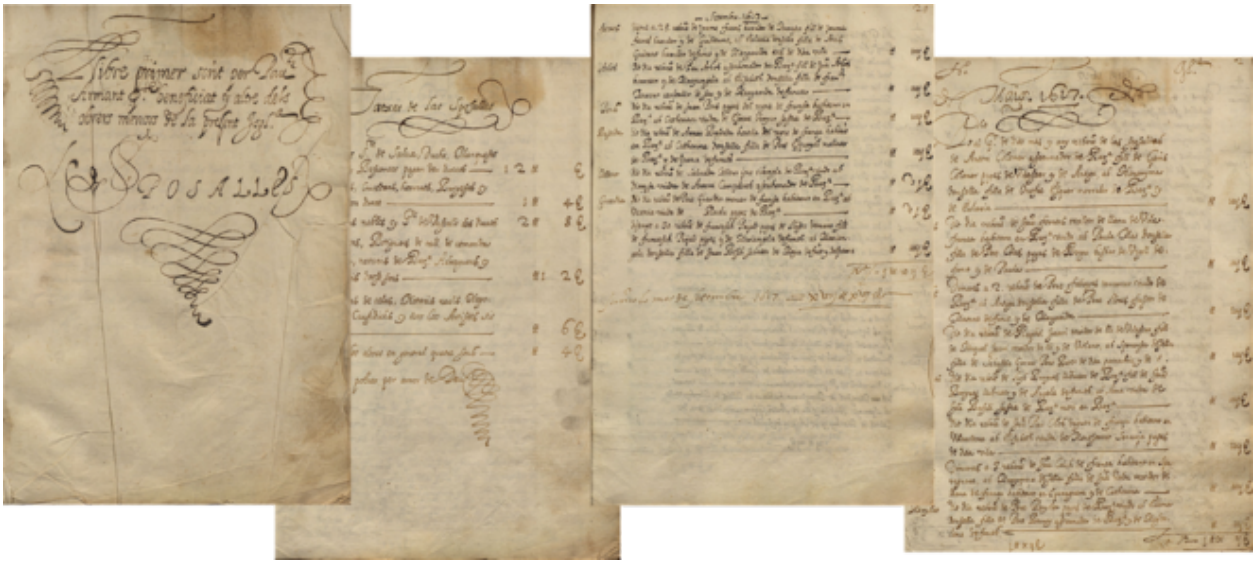


Figure 5.9: Examples of marriage records from different centuries.

WED is a typical example of record-structured page layout. A page (and each of the records it contains) is horizontally divided into three vertical zones, the *husband surname's zone*, the *main zone*, and the *fee zone*. Each of these zones are in turn vertically arranged into individual license records. The husband's family name and the day of the license is written at the left in the husband surname's zone, the marriage license itself is located in the the main zone, and the paid fee is indicated at the right side in the fee zone (See Fig. 5.10). Table 5.3 shows the nine types of *Layout-Regions* defined for WED, based on the below detailed *Layout-Elements*:

- Line Body (`LB`): Region occupied by the main body of a normal handwritten text line that runs along the entire width of the text block it belongs to.
- Start Record Line (`RL`): Main body region of a generally indented line, often opening a paragraph.
- End Record (`ER`): Region containing the body of a (shorter) text line which closes a license record.
- End Record with Ruler (`ERWR`): End record line body region which is completed with a horizontal ruler padding the full text width.
- Inter Line (`IL`): Region spanning between two consecutive text lines, generally crossed by the ascenders and descenders of the adjacent text lines.
- Interline with added text (`ILAT`): Interline space between two text lines which contains some added text to complement or correct text of the above or below adjacent lines.
- Blank Space (`BS`): Large rectangular region of blank space usually found as top/bottom page margins.

- Header (HD): Top page region containing the body of a header text. It usually indicates the month and year for that page.
- Footer (FT): Bottom page region containing the body of text usually unrelated with the last previous line.
- Non-text region (NT): Stands for everything which does not belong to any of the other regions.

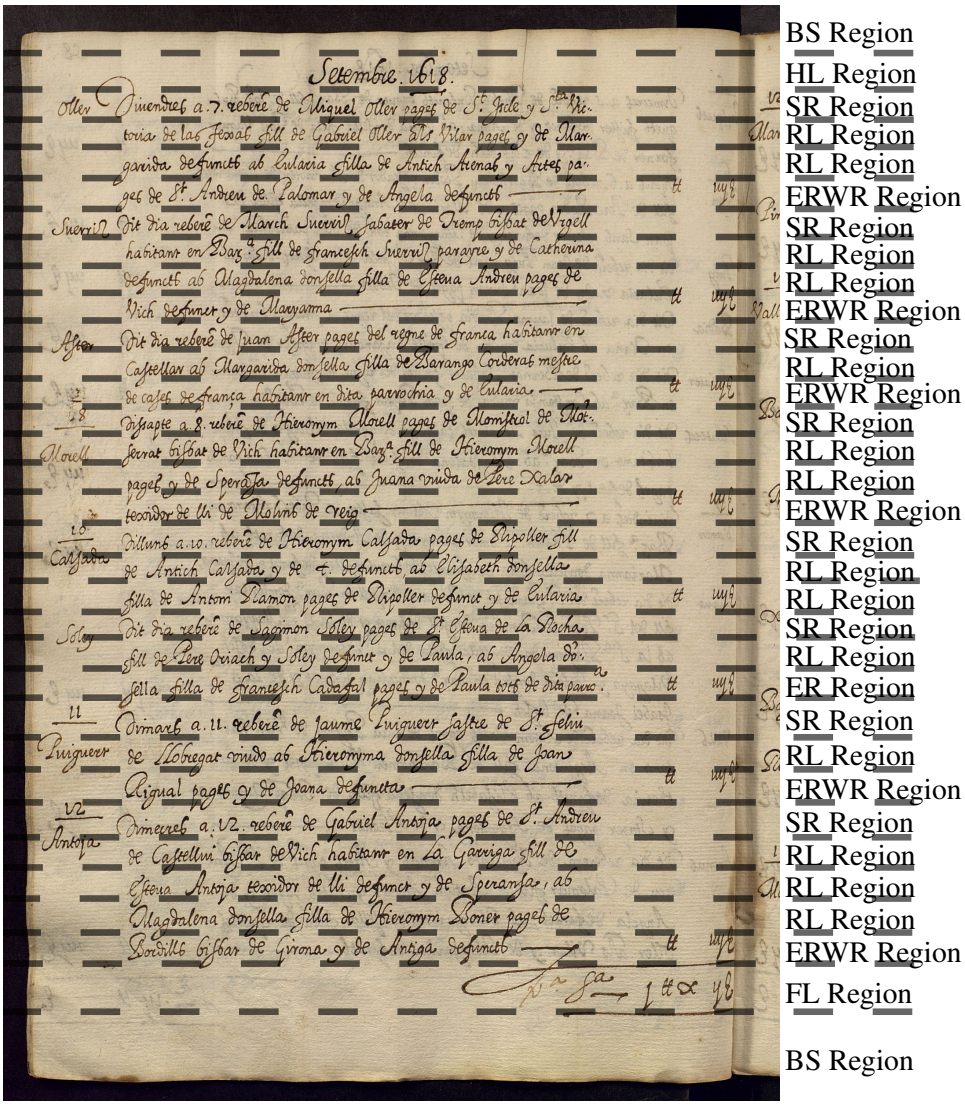


Figure 5.10: Page sample of the *Esposalles* corpus. The sample contains graphical annotations with the groundtruth generated for this corpus: the layout-regions and baselines corresponding to each of the different act region types and transitions

To create the groundtruth needed for evaluation, validation and test page images were processed in the same way as for the CHRIS corpus in order to obtain the reference baseline positions.

Table 5.3: LR dictionary and statistics of the WED partition used in this work ("*" in the last row stands for any LE).

Number of:	Train	Devel	Test	Total
Pages	100	50	23	173
Total text lines regions	3 361	1 650	794	5 804
REGULAR LINES (LB+IL)	1 144	550	239	1 933
START-RECORD LINES (SR+IL)	989	497	255	1 741
END-RECORD LINES (ER+IL)	130	68	38	236
END-RECORD WITH RULER (ERWR+IL)	854	429	211	1 494
HEADER LINES (HD+IL)	100	50	23	173
FOOTER LINES (FT+IL)	100	50	23	173
BLANK SPACES (BS)	200	100	46	346
NON-TEXT LINES (NT+IL)	0	0	0	0
INTER LINES WITH TEXT (*+ILAT)	44	6	5	55

5.7 Plantas

The PLANTAS collection was written employing quill-pen in the 17th century by the Spanish botanist Bernardo de Cienfuegos and is currently held at the Biblioteca Nacional de España [d]. The manuscript itself is composed of seven volumes of different number of pages. This corpus is part of the collection of corpora that are being groundtruthed as part of the TRANSCRIPTORIUM project.

For our experimentation we used the first volume of PLANTAS, entitled "Historia de las plantas". The volume has approximately 1 000 pages partitioned in the following manner:

- The first 49 pages contain: indices, reference tables and a botanical glossary in different languages.
- A 38-page preface written by Cienfuegos himself.
- 887 numbered pages divided into 152 chapters. These chapters contain information of different cereals and related plants, including 126 botanical illustrations.

This first volume contains about 20,000 handwritten text lines. The volume was digitized at 300 dpi in 24 bit RGB colour and saved as TIFF images by using an i2S SuprascanII scanner with the software Digibook. In addition, it employs a large number of Latin names, frequently used to identify the different plant species, and other non-Latin terms as: Greek, Arabic, Hebrew, Portuguese, Valencian, Catalan, French, German, English, Flemish, Polish, Bohemian and Hun. Example pages of this first volume can be seen in Fig. 5.11.

[d] Digital images available at Biblioteca Digital Hispánica: www.bne.es

Figure 5.11: Examples of PLANTAS page images from the volume considered.

Pages have rather wide margins, which often include margin notes, as well as vertical white space at the top and the bottom of each page (see Fig.5.11). Only simple layout analysis was needed to extract the relevant text boxes where the proposed TLAD approach was applied.

While the whole volume was used during out experimentation a more detailed evaluation was performed on the chapter called "PROLOGO". It consists of the first 38 pages of the first volume of PLANTAS with about 1200 text lines.

This corpus was considered to evaluate the text line detection capabilities in a production scenario. For this reason we considered only four types of *Layout-Elements*:

- **Line Body** (`LB`): Region occupied by the main body of a handwritten text line. In this case, no type of differentiation was made for type, indentation or length of the text line since this corpus was intended for text line detection evaluation.
- **Inter Line** (`IL`): Region spanning between two consecutive text lines, generally crossed by the ascenders and descenders of the adjacent text lines.

- **Blank Space** (BS): Large rectangular region of blank space usually found as top/bottom page margins.
- **Non-text Space** (NT): Stands for everything which does not belong to any of the other regions. Was mainly used to be able to process the different plant diagrams found in the pages.

Some of the above listed *Layout-Elements* were considered in order to be able to process the pages with the different elements contained. Since this corpus was used in order to evaluate text line detection only the **text line region**, which is composed of (LB+IL), *Layout-Element* is considered the actual result. Despite this, the rest of *Layout-Regions* were annotated in order to ensure correct training and decoding. An example of an annotated PLANTAS page can be seen in Fig. 5.12.

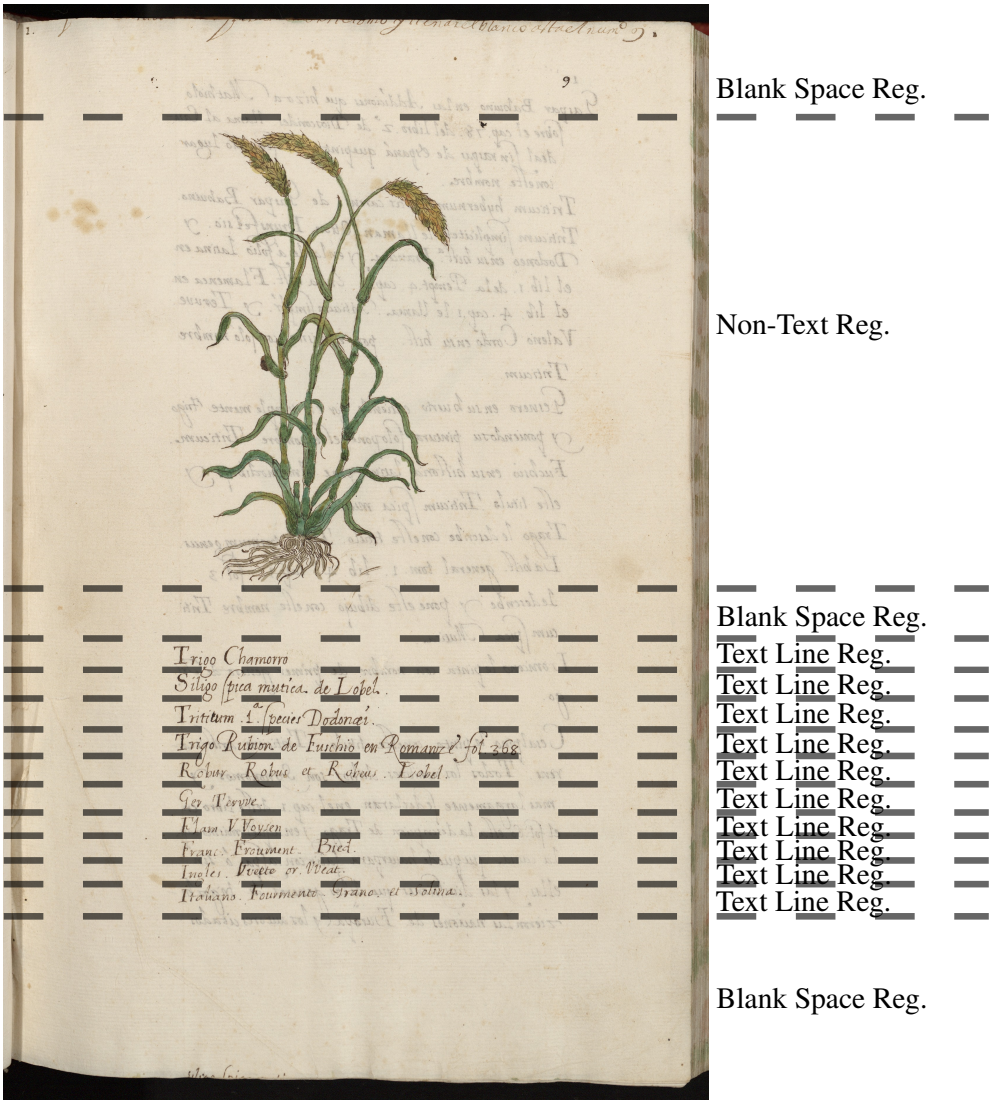


Figure 5.12: Page sample of PLANTAS corpus. The sample contains graphical annotations with the groundtruth generated for this corpus.

The 38 pages of the PROLOGO chapter of PLANTAS were divided into nine consecutive blocks in order to evaluate performance gains through various iterations. Table 5.4 shows some details of the these blocks.

Table 5.4: Statistics of the number of pages and lines in the PROLOGO chapter of the PLANTAS corpus. Data provided for each of the 9 blocks the 38 pages were divided into.

BLK-ID	Pages		Num. Lines
	IDs	Num.	
0	10	1	30
1	11-14	4	128
2	15-19	5	162
3	20-24	5	157
4	25-29	5	160
5	30-34	5	159
6	35-39	5	159
7	40-44	5	157
8	45-47	3	94

The rest of the PLANTAS volume was divided into 11 batches in addition to the Prologue (which is considered "batch 0", for initial system training) and was used for a more coarse longitudinal study. Statistical data of the volume can be seen in Table 5.5.

Table 5.5: Statistics of the number of pages in each of the blocks considered for the longitudinal evaluation with PLANTAS corpus.

Batch	No. Chapters	No. Pages
1	16	73
2	16	77
3	16	73
4	12	72
5	12	66
6	17	84
7	13	80
8	11	77
9	12	80
10	10	82
11	16	90
Total	151	854

5.8 RSEAPV

The *"Real Sociedad Económica de Amigos del País de Valencia"* (RSEAPV) (Royal Economic Partnership of Friends of the Country of Valencia) was an association of sorts established in 1776 by the Spanish King Carlos III. This partnership was, from its beginnings, a reference center for the whole of Valencian society, specially during the 18th century. The RSEAPV worked as as a focus group were most of the crucial issues were discussed and treated as they came to be.

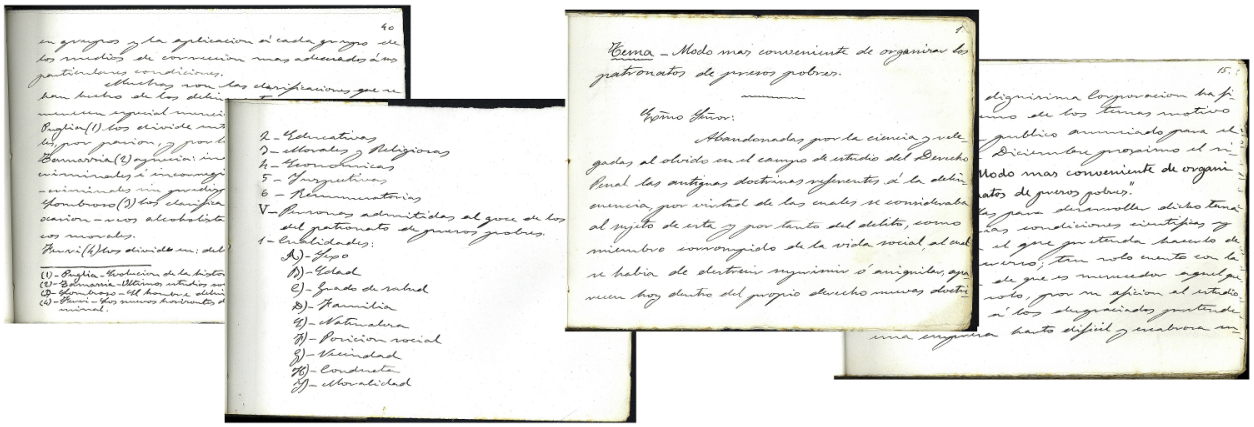


Figure 5.13: Examples from the RSEAPV database.

The RSEAPV has an archive that contains over 8,000 documents. These documents explain the history of the partnership from its inception to this present day. A large percentage of the collection, more than half, is composed by documents regarding the happenings in the Valencia during the 18th Century. These documents cover a myriad of topics in fields as diverse as arts, science, literature and economy. The archive, mostly written in the cursive style, has been lately digitalized and made accessible to the general public[e].

Although these documents are of great interest to researchers their automatic transcription is a difficult task due to the degradation problems, different handwriting styles, lexicon and notation complexity and heterogeneous layout structure.

From this vast collection we selected the text manuscript entitled *Report to award with the title "Modo más conveniente de organizar los Patronatos de los presos pobres", under the slogan "Via Vita, custodienti disciplinam: qui autem increpationes relinquit, errat" (cap. X. Vers. 17 Libro de los Proverbios)*[f]. The document was produced by a single writer in 1905. The contents written in Spanish span over 170 pages. In Fig. 5.13 we can see some page samples of this report.

Out of the 170 pages a set composed by the first 42 pages was selected to perform our experimentation on. This Corpus was used in order to review our text line detection performance as part

[e] https://riunet.upv.es/handle/10251/18484
[f] http://hdl.handle.net/10251/26346

of a end-to-end HTR system. For this reason we considered only four types of *Layout-Elements* as was performed for the PLANTAS corpus:

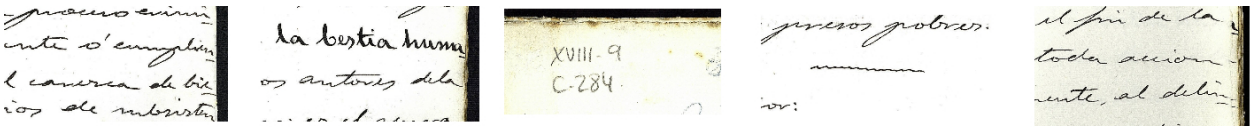
- **Line Body** (`LB`): Region occupied by the main body of a handwritten text line. In this case, no type of differentiation was made for type, indentation or length of the text line since this corpus was intended for text line detection evaluation.
- **Inter Line** (`IL`): Region spanning between two consecutive text lines, generally crossed by the ascenders and descenders of the adjacent text lines.
- **Blank Space** (`BS`): Large rectangular region of blank space usually found as top/bottom page margins.
- **Non-text Space** (`NT`): Stands for everything which does not belong to any of the other regions. Was mainly used to be able to process the different plant diagrams found in the pages.

Table 5.6 summarizes the basic statistics of the dataset text transcriptions.

Table 5.6: Basic statistics of RSEAPV dataset.

Number of:	Total
Pages	42
Lines	651
Running words	4,573
Lexicon size	1,497
Running characters	25,176
Character set size	80

As previously commented, the document presents some difficulties that complicate the layout process. In Fig. 5.14 we showcase some of the difficulties present in the RSEAPV corpus: a) hyphenated words and warping at the end of the line; b) different writing styles; c) degraded text; d) artifacts in the middle of the page; f) some part of the next page are shown.



a) b) c) d) e)

Figure 5.14: Examples of the difficulties in the RSEAPV database: a)hyphenated words and warping at the end of the line; b) different writing styles; c) degraded text; d) artifacts in the middle of the page; e) some part of the next page are shown.

The dataset was divided into two different partitions to perform the experiments. The initial partition consisting of the first 22 pages, was defined as the training partition. The remaining 20

pages composed the test partition on which the evaluation is to be performed. In Table 5.7 we present the basic statistics of the two blocks.

Table 5.7: Basic statistics of RSEAPV dataset.

Number of:	Train	Test	Cross-Val
Pages	22	20	5.25
Lines	303	348	81.4
Running words	2,150	2,439	572
Lexicon size	838	936	299.6
OOV	–	813	143
Character set size	67	77	76.25

5.9 Capitán

CAPITÁN is a huge archive of manuscripts of Spanish and Latin American music from the 16th to 18th centuries. These manuscripts were written using the so-called *white mensural notation*, which in many aspects differs from the modern Western musical notation. Furthermore, this archive was written following the slightly different Hispanic notation of that time, increasing its historical and musicological interest. The CAPITÁN archive is managed by the Department of Musicology of the Spanish National Research Council of Barcelona, which kindly allowed the use of the archive for research purposes.

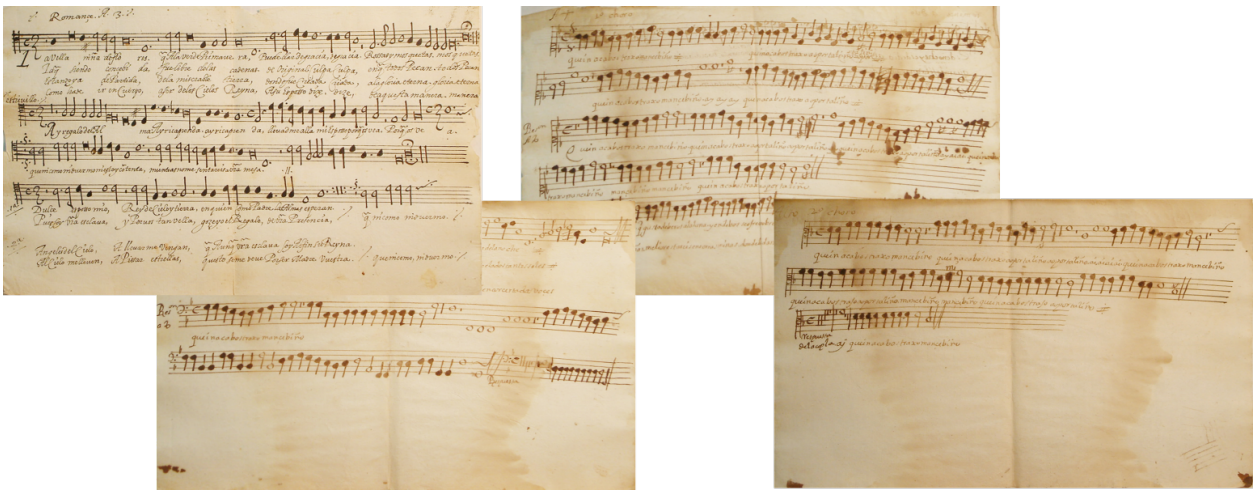


Figure 5.15: Example of pages of the selected music book from the CAPITÁN.

Examples of pages from this book are illustrated in Fig. 5.15. For our experimentation, 50 pages were arbitrarily selected for training and 46 for testing. Table 5.8 presents basic statistics of this dataset.

The page images of the archive considered in this work may contain up to five main types *Layout-Elements* to form the 9 types of *Layout-Regions*:

- **Title Line (TL)**: title of the piece that might appear at the beginning of a piece (top of the first page). It is not present in every page but only in those that represent the beginning of a music piece. It always appears at the top of the document.
- **Staff lines (SL, SL-A, SL-D, SL-DA)**: represent those regions which contain a pentagram. We have also considered subclasses of this region type in order to distinguish normal staffs (SL) from those that present many descending notes (SL-D), many ascending notes (SL-A) or both (SL-AD). The main interest of performing this differentiation between normal staff lines and the other sub-types is in the possible benefits this type of information might have on the actual note recognition.
- **Empty Staff Line (ESL)**: empty staves without musical content. Important to be differentiated as they do not require accompanying lyrics and they can not be transcribed.
- **Lyrics lines (LL, SLL)**: words that are sung appear below their corresponding staff. Sub classes have been created in order to distinguish normal Lyric Lines (LL) from Short Lyric Lines (SSL) that due not span the whole line because of the use of repetition symbols.
- **Blank space (BS, EBS)**: page regions in which there is no content. Given the difference in size and location, we have distinguished between those used between staves (BS) from those that appear at the end of a page (EBS).

In Table 5.8 we can see how the different *Layout-Elements* are combined to form each of the required *Layout-Regions* for this corpus. It also hold the information of how they are distributed in each of the different sets used. Examples of the different *Layout-Regions* accompanied with the visual segmentation of each region can be seen in Fig. 5.16.

Table 5.8: Layout regions dictionary and corresponding statistics of the CAPITÁN training and test sets used in this work.

Number of:	Train	Test	Total
Pages	50	46	96
Total text line regions	300	276	576
Total pentagram regions	300	276	576
Title Lines (TL+IL)	5	4	9
Staff Lines (SL+IL)	140	164	304
” with ascending notes (SL-A+IL)	143	79	222
” with descending notes (SL-D+IL)	6	10	16
Empty Staff Lines (ESL+IL)	23	11	34
Lyric Lines (LL+IL)	289	253	542
Short Lyric Lines (SLL+IL)	3	7	10
Blank Spaces (BS)	97	72	172
End Blank Spaces (EBS)	50	46	96

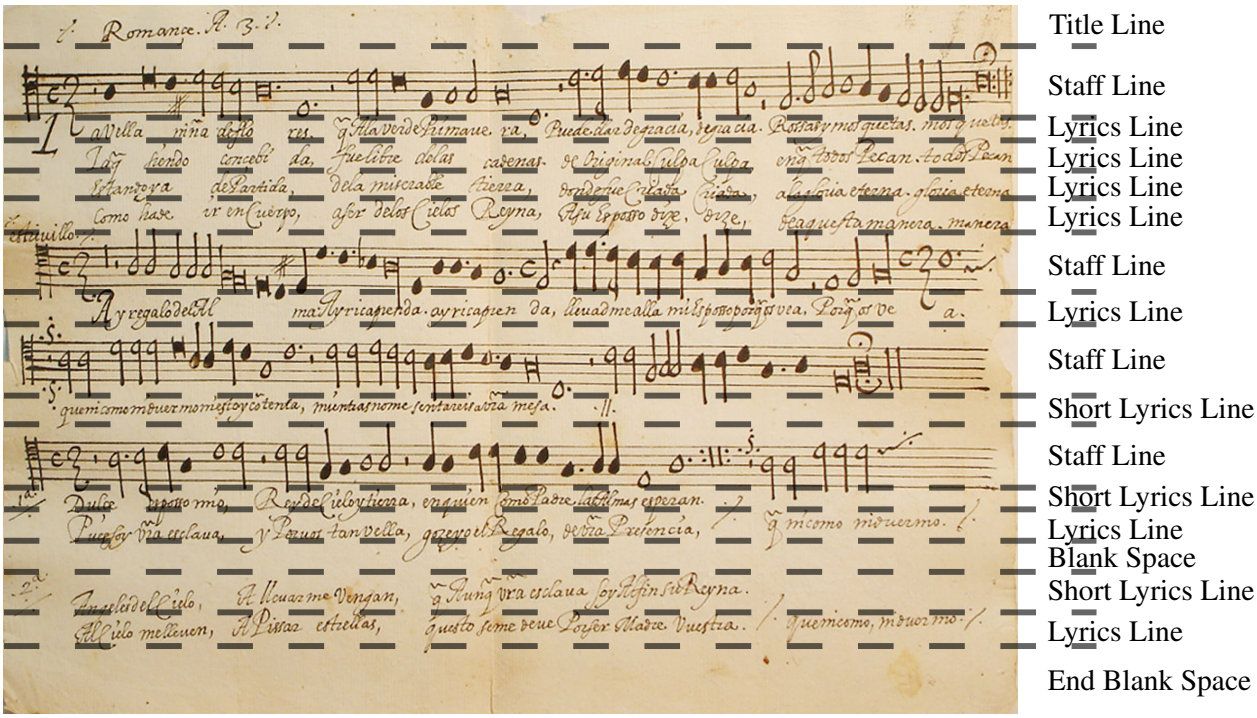


Figure 5.16: Page sample of *Capitan* corpus. The sample contains graphical annotations with the groundtruth generated for this corpus: the layout-regions and baselines corresponding to each of the title staff and lyrics regions

5.10 Chancery

The complete corpus named *Chancery* encompasses about 70 000 images reproducing the collection of medieval registers. The registers were produced by the French Royal Chancery (Paris, Archives Nationales, JJ 35 -JJ 211), in the period spanning from 1302 to 1483. They contain 68,000 charters given by the king of France. This large and iconic collection bears witness to the rationalization of late medieval administration. It is considered a key source to help our understanding of medieval Europe and the rise of centralized nation state. This rise appearing throughout the continent as a consequence of the long lasting wars between France and England.

Figure 5.17: Example of pages of the *Chancery* corpus

The sheer size of this corpus prevents scholars to study it as exhaustively as it deserves. A user-friendly access to the contents of this key resource would help increasing the knowledge about medieval history and promote ongoing research in comparative studies on state management and administration.

From the layout perspective this corpus is functionally divided into *Acts* or *Charters*. In these *Acts*, the current sovereign dictated statements on the creation of different organizational bodies and their rights and privileges were defined.

Acts can be large and distributed among different pages. Thus it is important for researchers to understand when it starts and finishes. If *Acts* can be correctly delimited researchers can look for specific charters that contain information on a specific topic.

In order to study how to perform such delimitation in the page images of the different batches the following *Layout-Elements* were considered:

- **Act Intro (I)**: Start of an Act text region that contains the typical repeated information and formula of a charter (name of the king, date, etc). Usually composed by a range (1-3) of the initial lines.

- **Act Middle (M)**: Middle part of an Act text region. It contains more generic words and unspecific text formula.
- **Act Final (F)**: Final part of an Act text region. Containing the the ending remarks of a charter and usual text formula for concluding one.
- **Act Transition Space (C,O)**: Page space between to acts. This space can be totally clear (C) leaving a nice visual queue for the start of a new act. But this transition space can be partially overlapped with text (O) due to the inclusion of dates, signatures or due to lack of space.
- **Blank Space (BP,EP)**: page regions in which there is nearly no content. Given the difference in location and content, we have distinguished between two types. The space at the beginning of a page (BP) that might contain some date or page number information. The end of a page blank space (EP) which is totally blank.

For our experimentation we considered two batches of the whole *Chancery* collection. This two batches were

First Batch: consisting in 56 images hand picked from the range of chapters from JJ038 to JJ091. The pages were hand picked by an expert in order to have a representative collection of the selected range of chapters. Details of this batch and the partitions used can be found in Tab. 5.9.

Table 5.9: Layout regions dictionary and corresponding statistics of the CHANCERY training and test sets for the first batch used during the Thesis experimentation.

Number of:	Train	Test	Total
Pages	30	26	56
Complete Act (I F | I M F)	21	17	38
Finishing Act (F | M F)	17	25	42
Starting Act (I | I M)	21	20	41
Medium Act (M)	1	1	2
Act Transitions (C | O)	33	36	69
Begin Page (BP)	30	26	56
End Page (EP)	30	26	56

Second Batch: consisting in 683 images with the full contents of the JJ078-JJ082 and JJ091 chapters. Details of this batch and the partitions used can be found in Tab. 5.10.

Table 5.10: Layout regions dictionary and corresponding statistics of the CHANCERY training and test sets for the second batch used during our experimentation work.

Number of:	Train	Test	Total
Pages	276	407	683
Complete Act (I F | I M F)	228	172	400
Finishing Act (F | M F)	107	196	303
Starting Act (I | I M)	121	203	324
Medium Act (M)	61	115	176
Act Transitions (C | O)	242	276	518
Begin Page (BP)	276	407	683
End Page (EP)	276	407	683

The two batches were fused and divided into a training, validation and test partitions:

- **Training**: A set of 176 images formed by joining the 56 images of the first batch plus 20 randomly selected pages from each chapter of the second batch.
- **Validation**: 96 images composed by selecting randomly 16 images from each chapter of the second batch after removing the images selected for the training partition.
- **Text**: composed of the 467 remaining images of the second batch after removing the images used by the training and validation partitions.

Examples of the *Layout-Regions* used in this corpus, for which the the statistics data was covered (Tab. 5.9 and Tab. 5.10), can be seen in Fig. 5.18.

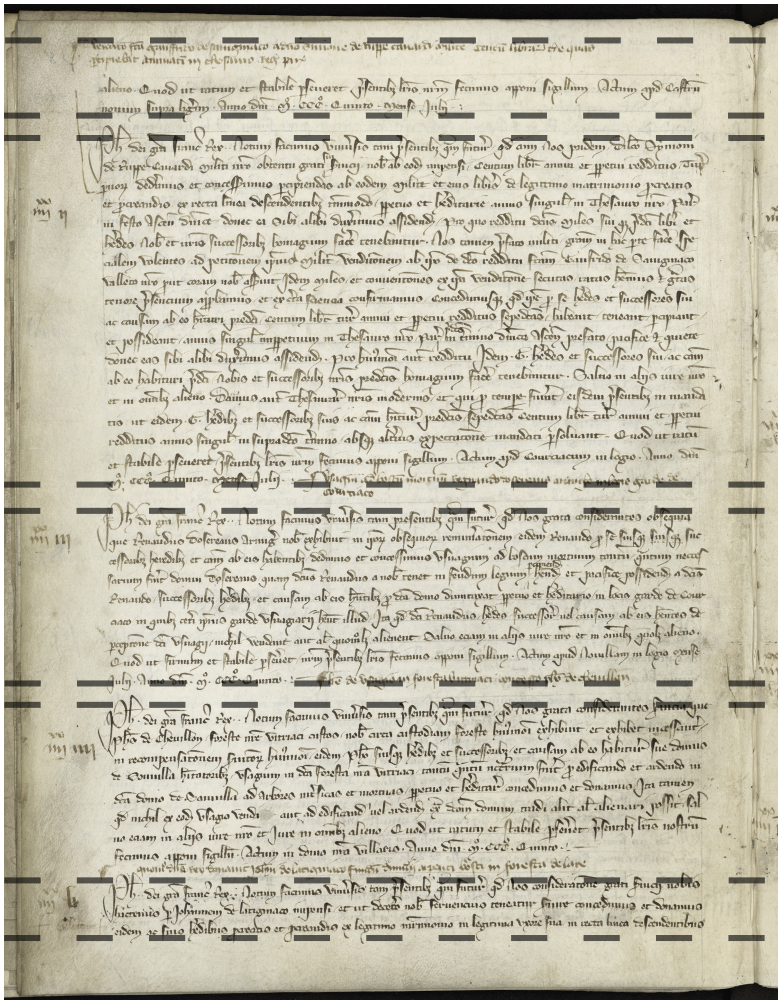


Figure 5.18: Page sample of the *Chancery* corpus. The sample contains graphical annotations with the groundtruth generated for this corpus: the layout-regions and baselines corresponding to each of the different act region types and transitions

5.11 Chapter Conclusions

In this chapter I have listed each of the corpus that will be used as the different experimental scenarios in which we will attempt to prove our scientific outcomes. For each corpus we have provided information regarding their main use, groundtruth available and partitions.

The corpus used in this thesis provide a heterogeneous collection of scenarios in which to experiment. The corpus cover different communication scenarios: records, prose, music, scientific while presenting the typical issues of historical handwritten text documents we described in Chapter 3 and are apparent in the different sample pages provided throughout this Chapter.

Bibliography

[1] Diem, M., Kleber, F., Fiel, S., Gruning, T., and Gatos, B. (2018). cbad: Icdar2017 competition on baseline detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1355–1360.

[2] Gatos, B., Antonacopoulos, A., and Stamatopoulos, N. (2007). Handwriting segmentation contest. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1284–1288.

[3] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2010). Icfhr 2010 handwriting segmentation contest. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 737–742.

[4] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2011). Icdar2009 handwriting segmentation contest. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1):25–33.

[5] Murdock, M., Reid, S., Hamilton, B., and Reese, J. (2015). Icdar 2015 competition on text line detection in historical documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1171–1175.

[6] Romero, V., Sánchez, J. A., Bosch, V., Depuydt, K., and de Does, J. (2015). Influence of text line segmentation in handwritten text recognition. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 536–540.

[7] Romero, V., Toselli, A. H., Rodríguez, L., and Vidal, E. (2007). Computer Assisted Transcription for Ancient Text Images. In *International Conference on Image Analysis and Recognition (ICIAR 2007)*, volume 4633 of *LNCS*, pages 1182–1193. Springer-Verlag, Montreal (Canada).

[8] Sánchez, J. A., Bosch, V., Romero, V., Depuydt, K., and de Does, J. (2014). Handwritten text recognition for historical documents in the transcriptorium project. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 111–117, New York, NY, USA. ACM.

[9] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., and Alaei, A. (2013). Icdar 2013 handwriting segmentation contest. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1402–1406.

CHAPTER 6

EVALUATION

Chapter Outline

6.1 Introduction

In this chapter we present the evaluation measures used to quantify the impact of the different approaches presented during the course of this thesis.

Since we cover a range of different problem scenarios were we apply different machine learning methods and a dynamic programming technique we must ensure that we use measures that are meaningful and measure correctly the quality of the yielded hypothesis.

6.2 Motivation

In order to understand the evaluation measures used, during the research work presented in this thesis, we must review the historical context in which it was written. At the time this doctoral dissertation was started (2013) there were no evaluation measures widely in use for region and text line detection. The community in fact did not consider *Text Line Detection* as an actual independent task but directly measured *Text Line Segmentation* by means of a Graphical Extraction Evaluation Tool we cover later in Section 6.8 of this chapter.

From the start we considered detection of text lines (or larger regions) to be a fundamental task and actually the most crucial part of any *Text Line Segmentation Approach*. Since at the time (2013) no accuracy measure for baseline localization existed we took it upon ourselves to define one as we explain in Section 6.6

Furthermore, one of the most significant contributions provided is the actual classification of such regions. Our novel approach to document layout via the use of Hidden Markov Models actually tackles directly the classification of regions. The actual localization of such regions is considered a by-product as we discussed in Chapter 3 . Due to the lack of measures on actual classification of regions we also had to define such an evaluation tool as we discuss in Section 6.5

As we do actually provide an approach for the *Text Line Extraction* problem we also used the Graphical Extraction Error tool covered in Section 6.8 in order to compare the results yielded by our technique to other systems.

During the course of the research work performed for this thesis other measures appeared in different competitions [8, 11]. Although these measures have some nice characteristics we consider that the need of thresholds in both of them in order to perform the calculation is a negative trait.

The *Transkribus Baseline Evaluation Scheme* [8] was actually developed as part of the *READ* project of which we have been part. This measure actually suffered some modifications throughout its different versions. We will describe in this chapter this evaluation measure its characteristics and how it compares to our developed location error measure *Relative Geometric Error*.

In any case, we consider that our chosen evaluation measures must adhere to the following three principles:

- Meaningful: the measure makes sense to humans that usually review the results yielded by such document layout systems.
- Adequate: the measure adequately reflects:
  - The time it would take a human to correct it.
  - Given a system that depends on the yielded results: improvements on the value of the evaluation measure should imply positive impact on the overall performance of that system.
- Efficient: we are able to launch it systematically and with little time/money impact. This is required in order to review improvements during development phases of a technique.

The aforementioned factors might seem obvious but given the issues found in highly utilized evaluation measures in other fields [2, 9, 18] and ours [15] we consider it crucial to establish how each measure is compliant with both the described qualities.

6.3 User Review Time

Given an specific hypothesis yielded by a system (or a human user) when reviewed by an expert time will be required in order to read, analyses and correct it if need be. The above mentioned time is named by us as User Review Time (URT).

This time required for revision can be divided into three different aspects:

Read time: Time required by reviewer to go through each of they yielded results for a specific document page.

Analysis time: Time required by reviewer to understand if a part of one the hypothesis (region label, baseline, extraction polygon) of the page has an issue or not.

Correction time: Time required by reviewer to perform corrections on the artefacts used to express the hypothesis.

This measure is the most basic way to review the performance of a given system. Although meaningful, and adequate it is not efficient due to the human involvement required to perform the calculation. Furthermore, the measure values obtained can not be guaranteed to be reproducible

when using different experts or with the same expert at different moments. Despite its shortcomings, this type of evaluation is actually part of our initial list of principles of what is required in order to decide on or create a new evaluation measure.

Although necessary most papers or competitions do not present a study on the relation between their designed evaluation measure and the user review time. This leads to the false idea that the tool adequately measures the quality of the yielded hypothesis. Furthermore it usually ends up not being trusted by the human experts that have to deal with the system results.

When considering user review time we must also understand the impact the chosen way to represent the result (labels, baselines, polygon,...) has on it. The more complex the artefact chosen to represent the result the more time it will take to review and correct.

6.4 Evaluation via Assessment of Impact in Higher Level Tasks

Assessment through the time taken by users to review the yielded result might not always be entirely adequate or a realistic scenario. This is the case for systems whose yielded output is generally intended as the input of another system.

Full document layout detection and *Text Line Classification* can provide additional information to the human experts interested in reading the documents contents. Regardless of this, *Text Line Segmentation* is mainly envisioned as a necessary task in order to perform higher level order tasks. The lines extracted provide minimal benefits to the human experts by themselves. These lines can generally be considered a *nice* by-product for user interface and alignment of the transcription with the image text uses.

Currently most *Handwritten Text Recognition* [7, 13], *Key Word Spotting* [1, 14], and *Text-Image Alignment systems* are dependent on *Text Line Segmentation*. This fact can be observed at their respective competitions [12, 16].

Hence evaluation measures for *Text Line Segmentation* should be related to the impact they produce on the higher level tasks that are dependent on it, as we discussed in Sec. 6.2.

For this reason we will also try to ensure a relation between our evaluation measures and how they impact on higher level *Handwritten Text Recognition* related tasks. Ideally we should measure how the variations in performance as per our chosen measurement tools relate/impact the performance of a system that uses the text lines as input.

In this case the study should be regarding the impact on a real *Handwritten Text Recognition* system. Thus we should stablish a relation between our evaluation measures and the Word Error Rate (WER) and the Character Error Rate (CER) [10]. Defined as the ratio between the minimum number of words/characters that need to be substituted, deleted or inserted to convert the sentences

recognized by the system into the reference transcriptions and the total number of words/characters in these transcriptions.

6.5 Region Error Rate

In order to assess the quality of the classification performed on the different line region elements we created the *Region Error Rate* (RER). Creation of the RER was necessary as at the time there was no measure in order to evaluate the classification of the different vertical elements that composed a region. In fact at the time the research, classification of the vertical components that composed the layout of a page, was not being tackled by the community.

Region Error Rate (RER) is calculated as the number of incorrectly assigned line region labels divided by the total correct line regions. The RER is obtained by comparing the sequences of automatically obtained region labels with the corresponding sequences. This is computed in the same way as the well known *Word Error Rate* (WER), with equal editing-costs assigned to deletions, insertions and substitutions [10]. This measure was created to provide insight on the quality of the classification.

As discussed in Chapter 7 the technology developed as part of this thesis to apply Statistical framework to document layout analysis can be applied at different levels. Hence we can train our system to detect higher/grosser level regions (paragraphs, diagrams, titles, foot notes, etc.) to a more lower/detailed text line level type of regions (start paragraph text line, end paragraph text line, signature text line, date text line, etc.) Our RER measure can adapt to these changes on the detail level without any issues.

Furthermore, independently of the chosen level of detail of regions, two different flavours of this measure can be considered depending on the intention of our assessment.

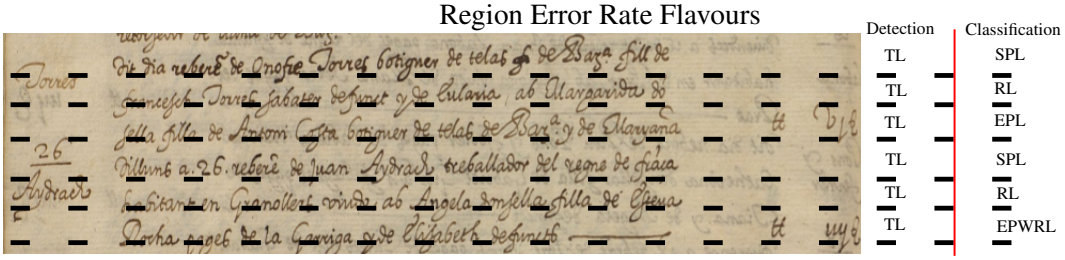


Figure 6.1: Diagram presenting the different labels that are to be detected in the Detection or Classification flavours of Region Error Rate for the same text segment of the Esposalles corpus (Sec. 5.6)

6.5.1 Detection Region Error Rate

Detection Region Error Rate (D-RER) is intended to assess only the presence or not of regions, regardless of their class labels (and, of course, excluding BLANK SPACES).

For example in Fig. 6.1 we can see the *Detection Region Error Rate* being applied to the Esposalles Corpus (Sec. 5.6). In this case the measure is being applied for a system trained for text line level type regions. As we can see in this situation the evaluation measure assesses the percentage of incorrectly detected text lines.

6.5.2 Classification Region Error Rate

For each corpus the regions of interest to be detected or classified will be determined by the needs of the end use. Furthermore those regions might be required to be classified as per a specific set of labels. *Classification Region Error Rate* (C-RER) evaluates the misclassification rate between the detected region types and their reference labels.

If we look again at Fig. 6.1 we can see how the different text line regions should be classified as per the chosen set of labels (Sec. 5.6). Any mismatch between the yielded hypothesis from the trained system will be considered an error.

In Fig. 6.2 we can see an example groundtruth segmentation and classification of regions (in this case called Acts) for the Chancery corpus (Section. 5.10). Any deviation from the specified class labels for each region would impact the C-RER.

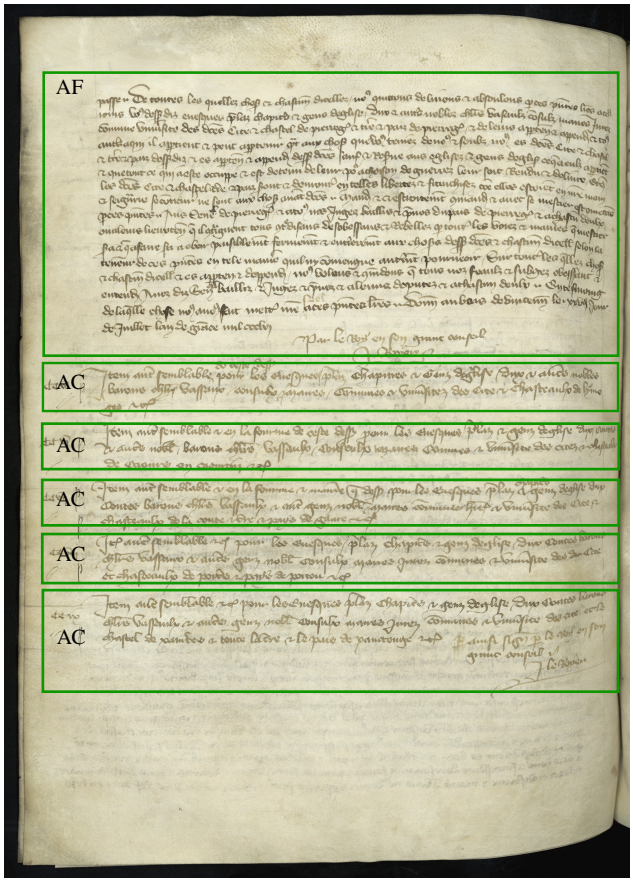


Figure 6.2: Picture of a sample Chancery corpus page(Section. 5.10) with the groundtruth region segmentation and labelling overlayed

6.6 Relative Geometric Error

The *Relative Geometric Error* (RGE) evaluation measure is calculated in three phases. First, for each page, we find the best alignment between the system-proposed baseline coordinates ($\hat{\mathbf{b}}$ in Eq. 3.8) and the page reference baseline coordinates ($\mathbf{r}$) by minimizing the accumulated absolute difference. This minimization can be performed by means of dynamic programming.

We define P as a possible alignment (list of operations) between two list of baselines ($\boldsymbol{b}$ and $\boldsymbol{r}$):

$$\hat{\boldsymbol{b}} = \langle \hat{b_1}, \hat{b_2}, \ldots, \hat{b_n} \rangle \tag{6.1}$$

$$\boldsymbol{r} = \langle r_1, r_2, \ldots, r_m \rangle \tag{6.2}$$

$$P = \{p_k = (\hat{b_i}, r_j) : \hat{b_i} \in \hat{\mathbf{b}} \wedge r_j \in \mathbf{r} \wedge 1 \leq k \leq n + m\} \tag{6.3}$$

We define the cost $c(k)$ of the alignment p_k as: $c(k) = c((\hat{b_i}, r_j)) = |b_i - r_j| \cdot w_k$. Hence we can define $W(P)$ as the weighted total cost of the sequence of operations in the following manner:

$$W(P) = \sum_{k=1}^{|P|} c(k) \tag{6.4}$$

where w_k is the specific weight factor associated to the alignment operation k, which takes the following values: 1 for insertion and deletion and 2 for substitution.

The alignment cost of two list of baselines **b** and **r** can be calculated by means of the normalized edit distance $d(\boldsymbol{b}, \boldsymbol{r})$:

$$d(\hat{\boldsymbol{b}}, \boldsymbol{r}) = \min_{P} \frac{W(P)}{L(P)} \tag{6.5}$$

where $L(P)$ is defined as the number of elementary weighted edit operations described by P. As we assume the cost of substitution to be double the cost of insertion or deletion $L(P)$ is actually a constant K, for all possible alignment paths. Thus, we ensure that substitution is not favored over insertion and deletion, therefore allowing us to simplify the last equation to:

$$d(\hat{\boldsymbol{b}}, \boldsymbol{r}) = \frac{1}{K} W(\hat{P}) \tag{6.6}$$

where $\hat{P} = \min_P W(P)$ which can be easily resolved by traditional dynamic programming technique.

Finally as we want the actual non-weighted difference between the reference and hypothesis coordinates we define the real cost $d_r(\boldsymbol{b}, \boldsymbol{r})$ as the non-weighted sum of differences of the minimum cost alignment $\hat{P}$.

$$d_r(\boldsymbol{b}, \boldsymbol{r}) = \sum_{k=1}^{|\hat{P}|} |b_{ik} - r_{jk}| : (b_{ik}, r_{jk}) \in \hat{P} \tag{6.7}$$

A graphical representation of an alignment cost calculation can be seen in figure 6.3.

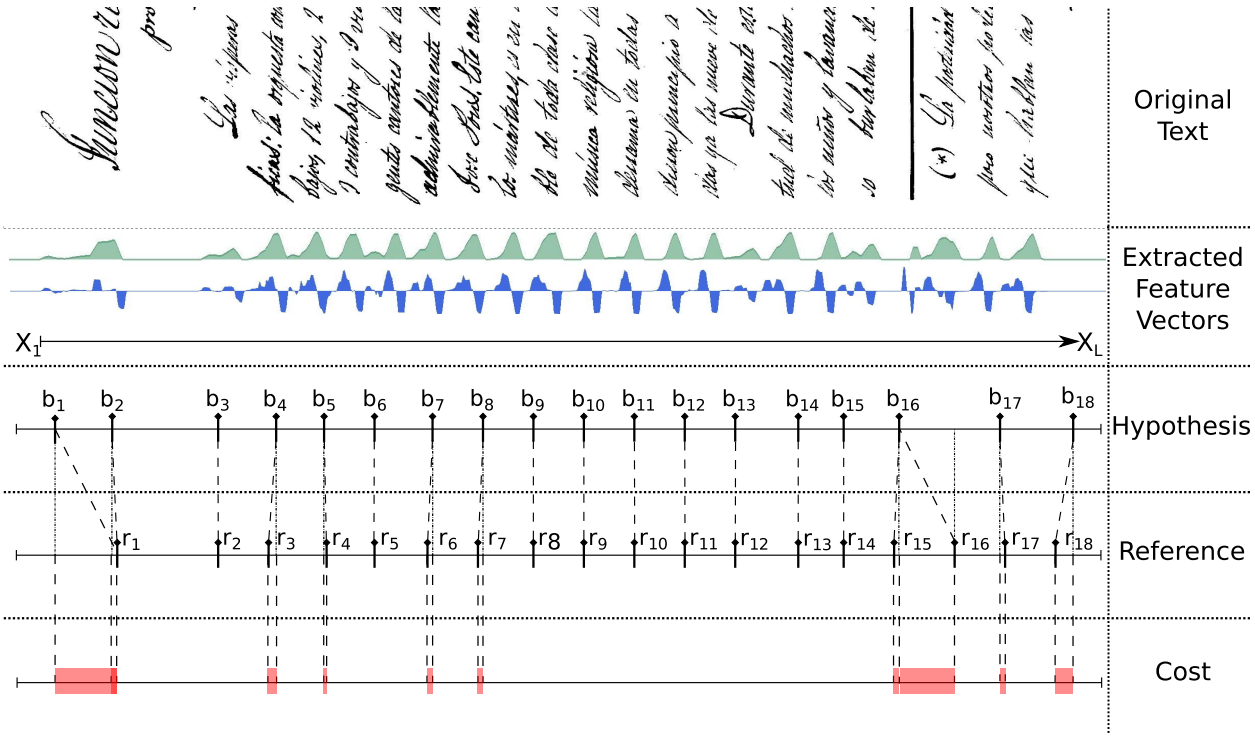


Figure 6.3: Illustration of the alignment cost calculation needed to determine the relative geometric error (RGE) of a handwritten text page. The cost is the accumulated horizontal length of the red shaded regions in the bottom line.

6.7 Transkribus BaseLine Evaluation Scheme (TBES)

This evaluation measure was first defined in 2017 as part of the *READ* project [8] and was initially used in an actual competition in the 2017 ICDAR congress [3]. The Transkribus baseline evaluation scheme (TBES) was defined due to the same issues we found with the existing available measures. In fact, the initial article defining the corpus and evaluation measure cites the work presented in this thesis [15] as one of the main reasons to create this new measure.

As we will be using this evaluation scheme, we will now perform a brief description of how the different values are calculated. We advise the reader of this thesis to read the original article [8] in case of doubt or need of additional information.

The **TBES** evaluation measure performs its calculations parting from annotated baselines represented as poly-lines. We will denote the groundtruth baselines set with the letter $\mathcal{G}$ and the hypothesis baselines set with the letter $\mathcal{H}$. These poly-lines are processed in order to have a normalized number of points so that the distance between each consecutive point is equal or less to $\sqrt{2}$.

These normalized chains are then used in order to calculate the three main values that define this evaluation scheme:

- **R-Value**: a value that tries to quantify how much of the existing text in the page is being detected, regardless of any hypothesis layout issues. This value has similar properties to the well known *recall*value but it is not exactly the same as we will later discuss.
- **P-Value**: measures how precisely the detected layout resembles the groundtruth. Also similar in some characteristics to the *precision* measure.
- **F-Value**: a harmonic mean between the **R-Value** and the **P-Value**.

The calculations of both the **R-Value** and **P-Value** are dependent on a *coverage* function $\text{COV}(p, q, t) \in [0, 1]$. This *coverage* function provides a Real value regarding to which degree are the points that define the baseline p *close enough* to another baseline q. Where a threshold t is used to determine when the distance exceeds what is considered adequate. This *coverage* function is extended $\text{COV}_S(p, \mathcal{Q}, t)$ so that the points of the baseline p are compared to a whole set of baselines $\mathcal{Q}$.

The tolerance chosen to perform this *coverage* calculation can greatly impact the results of the measures. In order to resolve this the evaluation scheme calculates a specific threshold t_g for each of the ground truth baselines $g \in \mathcal{G}$. These specific thresholds are calculated as 25% of the estimated interline space between the specific baseline being considered and its closest baseline. Given a set of groundtruth baselines $\mathcal{G}$, $\mathcal{T}(\mathcal{G})$ will denote the compiled set of t_g (or $\mathcal{T}$ for simplicity).

With the above defined COV_S and $\mathcal{T}$ the **R-Value** is defined as follows:

$$R(\mathcal{G}, \mathcal{H}, \mathcal{T}) = \frac{\sum_{g \in \mathcal{G}} \text{COV}_S(g, \mathcal{H}, t_g)}{|\mathcal{G}|} \tag{6.8}$$

It is important to note that as per this measure the text of a line is considered to be *recalled* if the groundtruth baseline g is covered by the whole set of hypothesis baselines $\mathcal{H}$. This measure will not penalize if a groundtruth baseline or text that is single unit is covered by various hypothesis baselines. This must be taken into consideration for certain technologies that could be impacted when text that is continuous is presented to the system, in the learning and/or decoding phases, in chunks.

Since the **P-Value** measure must evaluate the precision with which the provided layout defined by the hypothesis baselines $\mathcal{H}$ approximate the groundtruth layout defined by $\mathcal{G}$ some sort of best possible alignment must be performed between both sets. To resolve this, the creators of this evaluation scheme define the alignment set $\mathcal{M}(\mathcal{G}, \mathcal{H}) \subset \mathcal{G} \times \mathcal{H}$. Where $\mathcal{M}(\mathcal{G}, \mathcal{H})$ is constructed

with a greedy algorithm by taking pairs of baselines from $\mathcal{G}$ and $\mathcal{H}$ with maximal $\text{COV}(h, g, t_g)$ value.

$$P(\mathcal{G}, \mathcal{H}, \mathcal{T}) = \frac{\sum_{(g,h)\in\mathcal{M}(\mathcal{G},\mathcal{H})} \text{COV}_S(h, g, t_g)}{|\mathcal{H}|} \tag{6.9}$$

Extra baselines that do not align to any of the groundtruth baselines will penalize this result, as they will not increase the value of the numerator while the denominator stills considers them.

With the **R-Value** and **P-Value** defined can now define the **F-Value** measure:

$$F = \frac{2RP}{R + P} \tag{6.10}$$

This evaluation scheme resembles in certain ways our own **RGE** measure explained in the last Section. Both measures actually take into account the distance between the hypothesis and groundtruth baselines in their calculations. The **RGE** measure can be seen as equivalent to the **TBES** for text lines with little skew that can be annotated with simple straight baselines. Although the **TBES** scheme can measure errors in more scenarios than the **RGE** the use of a threshold in the *coverage* function might create some precision issues. In Chapter 7 we will review this evaluation scheme and compare it in a more thorough manner to our **RGE** measure.

6.8 Graphical Extraction Error Evaluation

In order to estimate the accuracy of the text line extraction tools developed as part of this doctoral dissertation we will need an evaluation measure. This measure must not take into account the difference between a hypothesis extraction polygon and the groundtruth one. If we perform the evaluation in this manner then any extraction polygon that contains all the elements of the text line but is not exactly the same as the groundtruth one will be evaluated as *having* errors.

To avoid this we will use an evaluation measure that estimates the accuracy of the extraction polygon as per it contents. The more foreground elements of a specific text line are enclosed by an extraction polygon the more accurate it should be rated.

In our research we have used to evaluate this polygon extraction error a handwritten segmentation contest measure [4–6, 17]. This competition and its metric has had its continuity in the community for several years. Although the assessment tool was initially envisioned to measure the whole text line segmentation task as a whole the community has shifted to other metrics [3, 11]. This is due to the new found importance of text baseline detection and to the issues found in this measure [15].

In any case these competition metrics seem adequate for *Graphical Extraction Error Evaluation* but have some shortcomings as we will see later in Chapter 7. These metrics are based on the *Match Score* function (M_S). The M_S function is based on the number of matches of foreground pixels between the line hypothesis and the actual lines present in the ground truth. The $M_S(i,j)$ represents the matching results of the j-th ground truth region and the i-th result region and is defined as:

$$M_S(i,j) = \frac{|G_j \cap R_i \cap I|}{|(G_j \cup R_i) \cap I|} \tag{6.11}$$

Where I is the set of all the image points, G_j the set of all points inside the j-th ground truth region and R_i is the set of the points inside the i-th result region.

A *one-to-one* match between a result region i and a ground truth region j pair is only considered if M_S is equal or above a threshold that is set by the evaluator. In our experimentation this threshold was set to *95%* as indicated in ICDAR 2013 for line segmentation evaluation [17]. Once defined this basic evaluation concepts we proceed to define the detection rate D_R and the recognition accuracy R_A as follows:

$$D_R = \frac{o2o}{N} \tag{6.12}$$

$$R_A = \frac{O_2}{M} \tag{6.13}$$

were N and M are the number of ground truth and result regions, respectively, and $o2o$ is the number of one-to-one matches as per the defined threshold.

Finally, we define the performance metric *F-Measure* F_M, as the harmonic mean of the detection rate D_R and the recognition accuracy R_A:

$$F_M = \frac{2D_R R_A}{D_R + R_A} \tag{6.14}$$

6.9 Chapter Conclusions

During the course of this past chapter we have:

- Provided an in depth description of the current evaluation measures in the field of Document Layout Analysis.

- Discussed the principles that must be fulfilled by evaluation measures to be of actual value for research.
- Detailed the user and higher level system related measures that our evaluation measures should relate to.
- Defined the different evaluation measures that will be used through out our experimentation.

Bibliography

[1] Aldavert, D., Rusiñol, M., Toledo, R., and Lladós, J. (2015). A study of bag-of-visual-words representations for handwritten keyword spotting. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(3):223–234.

[2] Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 136–158, Stroudsburg, PA, USA. Association for Computational Linguistics.

[3] Diem, M., Kleber, F., Fiel, S., Gruning, T., and Gatos, B. (2018). cbad: Icdar2017 competition on baseline detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1355–1360.

[4] Gatos, B., Antonacopoulos, A., and Stamatopoulos, N. (2007). Handwriting segmentation contest. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1284–1288.

[5] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2010). Icfhr 2010 handwriting segmentation contest. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 737–742.

[6] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2011). Icdar2009 handwriting segmentation contest. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1):25–33.

[7] Granell, E., Romero, V., and MartÃnez-Hinarejos, C. D. (2016). An interactive approach with off-line and on-line handwritten text recognition combination for transcribing historical documents. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 269–274.

[8] Grüning, T., Labahn, R., Diem, M., Kleber, F., and Fiel, S. (2017). READ-BAD: A new dataset and evaluation scheme for baseline detection in archival documents. *CoRR*, abs/1705.03311.

[9] Lavie, A. and Agarwal, A. (2007). Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

[10] McCowan, I. A., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P., and Bourlard, H. (2004). On the use of information retrieval measures for speech recognition evaluation. Idiap-RR Idiap-RR-73-2004, IDIAP, Martigny, Switzerland.

[11] Murdock, M., Reid, S., Hamilton, B., and Reese, J. (2015). Icdar 2015 competition on text line detection in historical documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1171–1175.

[12] Pratikakis, I., Zagoris, K., Gatos, B., Puigcerver, J., Toselli, A. H., and Vidal, E. (2016). Icfhr2016 handwritten keyword spotting competition (h-kws 2016). In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 613–618.

[13] Puigcerver, J. (2017). Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 67–72.

[14] Puigcerver, J., Toselli, A. H., and Vidal, E. (2014). Word-graph and character-lattice combination for kws in handwritten documents. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 181–186.

[15] Romero, V., Sánchez, J. A., Bosch, V., Depuydt, K., and de Does, J. (2015). Influence of text line segmentation in handwritten text recognition. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 536–540.

[16] Sánchez, J. A., Romero, V., Toselli, A. H., Villegas, M., and Vidal, E. (2017). Icdar2017 competition on handwritten text recognition on the read dataset. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1383–1388.

[17] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., and Alaei, A. (2013). Icdar 2013 handwriting segmentation contest. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1402–1406.

[18] Vidal, E., Toselli, A. H., and Puigcerver, J. (2015). High performance query-by-example keyword spotting using query-by-string techniques. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 741–745.

CHAPTER 7

EXPERIMENTATION

Chapter Outline

7.1 Introduction

In this chapter we present empirical studies of how the methods that were defined in chapters 3 and 4 for HTS fared when used in the different scenarios. The scenarios, defined by the set of corpus described in Chapter 5, will help us flesh out the different characteristics of our proposed techniques and provide empirical results. From these results we can hopefully confirm or disprove our initial hypotheses.

7.2 Motivation

In order to be able to comment adequately on our hypothesis we must have empirical results to sustain our affirmations. To yield these empirical results we will formulate a series of experimental set-ups. The experimental set-ups will consist of applications of the framework described in previous chapters to scenarios described by the different off-line corpora presented in Chapter 5 which will be evaluated via the measures defined in Chapter 6.

Before we start to describe the different experimental set-ups and results obtained, it is important that we briefly recap on our hypotheses and indicate how we plan to provide results that will allow us to provide commentary on them.

As we indicated during Chapter 1, idea that was expanded on in later chapters, the classical focus of the document layout analysis community was on text line/region segmentation as a whole. As we discussed previously text region segmentation is a task that can be divided into detection and extraction. While the extraction of the text region is important we consider the actual heavy lifting of text segmentation to be performed by the detection task. This focus on text region detection and text region classification, that differs from what the community was doing at the time, forced us to define our on evaluation measures. Furthermore, as indicated in our hypotheses, we believe there is a disconnection between the text region/line extraction evaluation measures and the impact it actually produces on the users (humans or other systems) that consume the extracted text regions.

Additionally to our different focus on text region classification and detection, we also intend to alter how to perform these tasks. One of our hypothesis is that this tasks can be performed by means of stochastic frameworks and thus, move away from the traditionally used heuristic methods that we can see throughout the literature (Chapter 2). Our aim is to perform both tasks simultaneously as indicated in Chapter 3. We plan to demonstrate that this problem can be solved easily, for any given corpus, with a simple adaptation of the region dictionary and the vertical layout model. Furthermore, we will also apply our framework to several production scenarios in order to understand its performance, what is the entry cost for this technology and if it escalates adequately as more training data is made available.

As indicated in Chapter 3 our framework can be easily applied for regions of different scales. The method work both at text line level, with larger text regions and with mixes. We will show

this trait of our approach throughout the different experiments documented in this chapter. Furthermore we will also demonstrate some of the issues that arise when trying to perform text region classification with only graphical features. We plan to demonstrate the use of word probabilistic indexes in order to extract valid information that will enhance the performance of text region detection and classification. This use of automatically generated word probabilistic indexes is a novel idea that we believe will have a dramatic effect in tackling the region classification problem and other document layout analysis tasks.

7.3 Text Line Detection versus Text Line Extraction

As commented in Chapter 2 and in Chapter 3 the problem of Text Region Segmentation consists in the detection of the specified regions and their extraction. Text Line Segmentation is the particular sub-set of instances of Text Region Segmentation were the regions considered are at text line level.

Despite that the detection of the text lines is considered to be the most crucial phase of text line segmentation, there is no actual study regarding the impact on the final result each one has. This issue is magnified by the fact that the evaluation of the different segmentation approaches was classically performed at a graphical level as per the final extraction polygon (see Sec. 6.8).

In addition to all of the above, the community has now started to shift its focus and considers tackling the baseline detection problem in an independent manner [5]. This shift creates the need for methods that are able to calculate the polygonal extraction frontier based on the baselines yielded by the new text line detection systems.

Due to the aforementioned issues and change of focus in the community, we performed the following experimentation in order to asses our text line extraction technique (Chapter 4) and evaluate the importance each of the subtasks has on the final results of the text line segmentation.

7.3.1 Experimental Set-up

The set-up here described has a dual purpose. On one hand, we wish to measure the benefits provided by our text line extraction approaches described in Sec. 4. On the other, we wish to quantify how important is the actual computation of extraction polygon versus the position detection of the text line in a real segmentation task.

In order to accomplish our dual objective and compare the efficacy of our distance map based approach to already existing methods, we will use the Handwriting Segmentation Contest Corpus-2013 edition (see Sec. 5.3). All results yielded for this corpus were evaluated by means of the official competition corpus measure described in Sec. 6.8.

Since our methods required as input the text baselines provided by another system, we must evaluate how the quality of the text baselines impacts our method. This, in its turn, allows us to understand the impact the quality of the detected baselines has on the extraction algorithms.

To do this we evaluate our methods in two scenarios. The first scenario consists in using the groundtruth baselines that were made for the corpus as the input for our text line extraction approach. This will help us understand the potential of our approach when it is not hindered by issues created by the text line detection hypothesis provided.

In the second scenario, we measure how our method would perform in a more realistic scenario in order to make an adequate comparison to other methods. In this case we used non-reviewed hypothesis baselines as input. These automatic baselines were provided by a state of the art baseline detection method [6] based on extremely randomized trees and the dbscan algorithm. We will refer to this baseline detection method as PRHLT-17.

We also wanted to measure how important is the actual computation of the text line extraction polygon in comparison to the text baseline detection task. In order to do this we used the simple extraction method described in Sec. 4.2 as a reference method.

The basic reference method performs a straightforward projection from the already existing baselines in order to compute the extraction polygon. This simple approach, which just computes its area by taking some pixels above and below along the path of the baseline, can be considered as the simplest extraction that can be directly computed from a given baseline. Due to its simplicity we consider all merits of the final extraction polygon to rely on the quality of the baselines.

As per the actual technical set-up, the graphical error evaluation required nothing special. In order to train the extremely randomized trees required for the automatic text baseline detection method [6], used in the aforementioned second scenario, only five pages from the training partition in the case of the *ICDAR 2013 data set* were required.

7.3.2 Results: detection over extraction

Next we will review the results of our experimentation. We compare the results provided by the reference method and our proposed approach based on distance maps with the results of the ICDAR 2013 competition participants [17]. To this list we have added the results obtained by a more recent graph based approach [7] and by one of the latest results based on clustering [9]. The results can be seen in Table 7.1.

First of all, we will look at the results obtained by the basic polygonal projection method (highlighted in green). This reference method managed to obtain a 86.59% score with the automatically detected baselines. This result is superior to some of the competition results that actually have some kind of intelligent method in place to perform the extraction. As we are able to reach a 89.25% in F_M with the groundtruth baselines and the simple extraction method, this implies that

Table 7.1: Evaluation results using the ICDAR 2013 Database. Where M is the count of result elements, $o2o$ is the number of one-to-one matches, D_R is the Detection Rate, R_A is the Recognition Accuracy and F_M is the harmonic mean. The number of ground-truth elements N is 2649. Results sorted as per the F_M measure value. We highlight in colour the new results provided by us.

Method	M	$o2o$	$D_R(\%)$	$R_A(\%)$	$F_M(\%)$
REGIM	4563	1629	40.38	35.70	37.90
AegeanUniv	4054	3130	77.59	77.21	77.40
PRHLT-17 + Polygonal Projection	2848	2380	89.84	83.56	86.59
ETS	4033	3496	86.66	86.68	86.67
Jadavpur Univ	4075	3541	87.78	86.90	87.34
GT. Base lines + Polygonal Projection	2649	2364	89.27	89.24	89.25
LRDE	4423	3901	96.70	88.20	92.25
PPSL	4084	3792	94.00	92.85	93.42
PRHLT-17 + Proposed Method	2726	2726	95.8	93.10	94.43
PortoUniv	4028	3811	94.47	94.61	94.54
CASIA-MSTSeg	4049	3867	95.86	95.51	95.68
URO-17	2664	2563	96.75	96.21	96.48
CVC-14	4176	3971	98.40	95.00	96.67
CMM	4044	3975	98.54	98.29	98.42
PAIS	4031	3973	98.49	98.56	98.52
INMC	2650	2614	98.68	98.64	98.66
ILSP-LWSeg-09	4043	4000	99.16	98.94	99.05
GT. Baselines + Proposed Method	2648	2638	99.62	99.58	99.60

only 10% is left to be gained by a more intelligent extraction method. From this we can conclude that most of the heavy lifting required for the complete text line segmentation is actually performed by the actual detection and localization of the text lines.

Our text line extraction approach performs adequately in both test scenarios (highlighted in blue). In the first scenario, when using the groundtruth baselines, it is able to achieve a near-perfect score. When using the *PRHLT-17* [6] automatically detected baselines, our second scenario, our approach has an adequate performance. The issues present in the automatically detected baselines do impact the results yielded by our method as we can deduct from the lower accuracy results.

In both scenarios our proposed method outperforms the reference polygonal projection method by at least 7%.

7.3.3 Discussion

In this section we presented the results for our text line extraction method based on distance maps that described in chp. 4. This text line extraction approach is applicable to modern as well as

historical handwritten text. The graph based method, through the use of a distance map and a simple dynamic programming technique, is able to provide the optimal equidistant segmentation boundary between two adjacent text lines.

The method takes as input (built on) previously detected baselines that can be provided automatically by other systems or could even be used after the review of a human expert. As seen in the results our method is able to make use of the baseline data effectively. It provides results in accordance to the quality of the baselines provided. Reaching the level where it can yield near-perfect results when taking as input ground-truth quality baselines.

This experimentation has clarified the impact the extraction method has on the overall result of the text line segmentation task. The results obtained allow us to affirm that the detection subtask has the highest responsibility of the two in resolving the complexity of this problem.

7.4 Graphical Extraction Error versus Baseline Graphical Error

Traditionally, Text Line Segmentation has been evaluated by means of a graphical error measurement that only considers the final extracted region (Sec. 6.8). This evaluation measure disregards the importance that the detection phase has on the overall result of the segmentation. Not only does this measure not allow us to understand the issues that arise at the detection level, it also seems to have issues with how it relates to the evaluation measures of the systems that depend on the extracted text lines.

Due to the aforementioned issues and doubts, we have performed the following experimentation in order to clarify the relation between the classically used measure and the impact it has on a real Handwritten Text Recognition tasks. Ideally, the higher quality the segmented text line has as per the competition measure the better results should be yielded by the text recognition system.

Additionally, we have also evaluated how our baseline based graphical error measure **RGE** (described in Sec. 6.6) relates to the performance measurements performed in the same task. We have performed this evaluation in order to review if our evaluation measure is adequate (see Sec. 6.2) as per the system that consumes the yielded text line it measures.

7.4.1 Experimental Set-up

In order to perform this study we required a corpus for which we had: the groundtruth baselines, groundtruth extraction polygons and the aligned text for each of the text lines. The groundtruth baselines will be used in order to perform the text line segmentations. Once we have the text lines

are segmented together with the transcription text, we can successfully train the models used in a Handwritten Text Recognition (HTR) system and evaluate them.

The *C5 Hattem Manuscript* corpus described in Sec. 5.4 complies with the aforementioned prerequisites. This corpus was chosen due to the fact that not only does it the required groundtruth baselines, but it also provides them with different levels of detail. In order to obtain the required WER and CER results we performed an 8-block (of 5 pages each) cross-validation experiment with the text lines yielded by each combination of baselines and extraction technique used (Chapter 4).

For the experiments carried out here we used ground truth transcripts annotated with the abbreviation symbol used and also with the hypothesized expansion of the abbreviation. The line images of the training partitions were used to train corresponding character HMM models using the standard embedded Baum-Welch training algorithm [11]. A left-to-right HMM was trained for each of the elements appearing in the training text images, such as lowercase and uppercase letters, symbols, special abbreviations, possible spacing between words and characters, etc. Meta-parameters of the HTR feature extraction modules, as well as corresponding HMM models, were optimized through cross-validation experiments. The optimal number of states per HMM was 8 with 64 Gaussian densities per state.

In the scenario where the competition measure is used we will vary the quality of the extraction polygon provided for the segmentation. This variation will be provided by the different extraction methods described in Chapter 4. In this manner we will be able to register the graphical error of the extraction and how it relates to the final WER and CER results of the HTR experiment performed with the yielded text lines.

On the other scenario, where we use the baseline based evaluation measures, we will alter the quality of the baselines used while keeping the extraction method fixed. This change in quality of the baselines used will be provided synthetically by applying Gaussian noise to the vertical coordinate of each of the points that compose the baselines. By applying this noise with a Gaussian distribution centred on the baseline point with increasing levels of standard deviation, we will generate baselines of less quality. With this set-up, we will be able to study how the noise is measured by our evaluation measure **RGE** (see Sec. 6.8) and how our measure relates to the newly developed baseline based evaluation measure **TBES** (see Sec. 6.7) and to the final WER results of the HTR experiment.

7.4.2 Results: Ensuring the Adequacy of our Measure

First, we will evaluate how the extraction method and quality of the input baselines affect the automatic transcription of the text line images. We will study how that impact is captured by the competition graphical measure and the error measurement tools used in HTR. In Table 7.2 we present how the quality of the extraction polygon affects the WER and CER in this HTR task.

As we can see in the above results, our extraction approach outperforms, by at least 10%, the equivalent basic projection result in the competition graphical error. Although this was to be

Table 7.2: Comparison table of the results obtained with the different combinations of used baseline detail and extraction method. Results are compared to the best possible value obtained with the actual hand reviewed groundtruth extraction polygons.

Extraction Method	Groundtruth	Basic Projection		Dynamic Programming	
Baseline Type	NA	Straight	Line Segments	Straight	Line Segments
o2o	1592	1217	1306	1376	1405
F_M (%)	100	76.4	82.0	88.4	93.1
WER	34.8	36.3	35.4	37.83	35.18
CER	15.8	18.1	17.3	17.9	16.2

expected it is important to note that this does not translate into a significantly better text recognition rate. In fact, in the case of using the straight baselines as input, the lines extracted that are actually of higher quality as per the competition measure actually impact negatively the HTR system.

The most consistent positive impact on both the competition measure and the HTR experiment results, are the caused by the different quality of baselines. The higher the level of detail in the baseline the better results are obtained in both. We must also note that the lack of proportionality and relation between the graphical measure based on the extraction polygon and the evaluation measures used in HTR. Even so, the WER and CER results only present a minimal difference between the results calculated with the straight baselines and the ones yielded with the line segment baselines.

All of the above seems to indicate that HMMs are more susceptible to loss of parts of the graphemes than to noise caused by parts of other text lines invading the final text line image used for training/decoding. Hence, the idea that a smaller graphical error, as per the competition text line segmentation metric, will always yield better HTR results (or of other tasks) is not true for all cases.

Upon closer inspection of the extraction polygons provided by our dynamic programming approach we found that some of the issues where derived from the actual baselines used as input. In some cases the results yielded from the straight baselines were not optimal due to the search area not allowing the algorithm to pick the optimal result as we can see in Fig. 7.1. This adds to the idea that an actual measure that evaluates solely the text line detection is required.

Next we will study how the quality of the baselines affects the HTR results. In order to do so, as commented in the previous subsection, we will modify the vertical coordinate of the baseline points by means of synthetic noise. The noise will be modelled by means of a Gaussian distribution centered on the baseline point from which we will randomly obtain the new value of the coordinate. This synthetically generated variations of the baseline will be measured by means of our geometrical baseline error **RGE** and the **TBES** evaluation measures. With the generated baselines we will extract the text line images in order to perform our HTR experiments. The text line image extraction is performed via the simple basic extraction method (see Sec. 4.2) in order to ensure the issues created due to the baseline quality are transfered in the most pure manner to the HTR experiment.

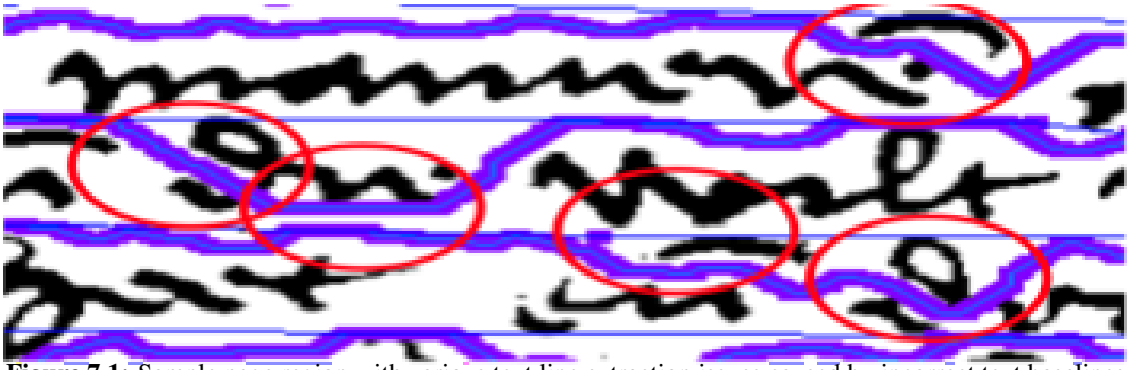


Figure 7.1: Sample page region with various text line extraction issues caused by incorrect text baselines.

This process will be performed with Gaussian distributions, used to generate the synthetic noise, of increasing standard deviation value and will be repeated five times for each distribution. With the described results we will study the relation between baseline quality and how it relates to the performance measurements of the HTR experiment. We will also be able to evaluate how our own measure **RGE**, developed in 2013, performs in comparison to the more modern evaluation scheme developed as part of the READ project **TBES**. In Table 7.3 we can see the results of this study.

Table 7.3: Study of how baseline quality is impacted by synthetically generated noise created by means of a Gaussian distribution. The impact is measured at the graphical level via three measures: an actual calculation of how much in average we randomly moved the points of a baseline, the **RGE** and **TBES**. We also measure the impact at the HTR level by performing an experimentation with the extracted lines. Each presented value is the average calculated over five independent executions performed for each Gaussian distribution.

Gaussian-Noise Dist. Standard Deviation (σ)	WER (%)	TBES ($1 - F_M$)	RGE (%)		Pixel Deviation (pix)	
			avg.	std-dev	avg.	std-dev
0	36.3	0	0	0	0	0
1	36.32	0	1.31	1.37	0.59	0.62
2	36.48	0	3.08	2.55	1.39	1.15
4	37.11	0.01	6.48	5.24	2.92	2.36
8	39.05	0.02	12.64	10.15	6.14	4.57
16	49.86	0.11	25.64	30.26	11.94	13.62
32	73.43	0.34	38.35	69.33	18.16	31.2

In the above table we can see how the **RGE** measure is able to correctly reflect the actual pixel deviations measured. As the value of the standard deviation of the Gaussian distribution we use to generate the noise increases the height difference between the groundtruth points and the newly generated points increase. This increase in deviation is correctly reflected by an increase of the error the **RGE** measures.

Furthermore, as the **RGE** error increases due to the increase of noise in the baselines the negative impact in the HTR system increases. A clear relation between our geometrical baseline error

measure **RGE** and the **WER** measure of the handwritten text recognition can be seen. In Fig. 7.2 we have represented graphically both measures as we increment quadratically the standard deviation of the noise distribution. The relation between both error curves is visible at plain sight. This relation between both measures can be more clearly seen the scatter plot represented in Fig. 7.3. The results for each of the different used Gaussian distributions create distinct clusters.

Contrary to how the **RGE** behaves the **TBES** measure does not seem to adequately register the discrepancies between the groundtruth baselines and the newly generated ones with synthetic error. We believe this occurs due to the *threshold* parameter of the **TBES** measure that is used in order to determine when a geometrical difference is to be considered an error or not. Standard executions of the **TBES** measure software allows it to estimate an adequate *threshold* parameter but our empirical results show that this estimation might not be entirely correct. The **TBES** results obtained for the baselines generated with the distributions of $\sigma \geq 4$ are specially worrying as the impact on the HTR accuracy does not seem reflected in them.

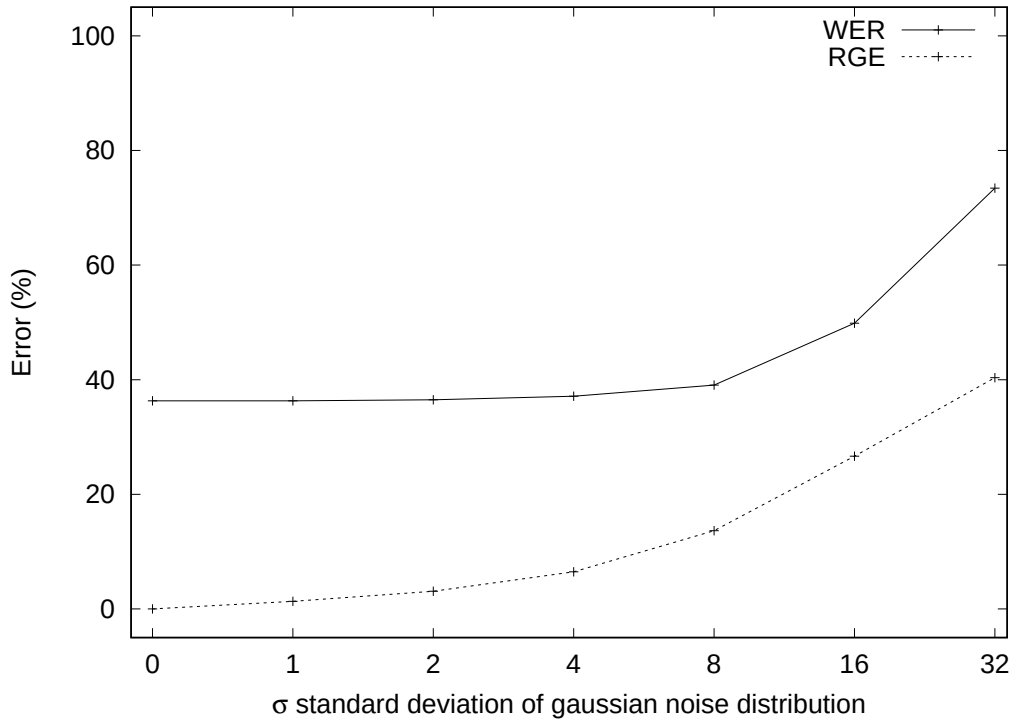


Figure 7.2: Figure shows the WER and RGE error curves (in logarithmic scale) as a function of the standard deviation of the Gaussian distribution used to generate the baselines. The represented value is the average for each measure calculated over the individual repetitions results.

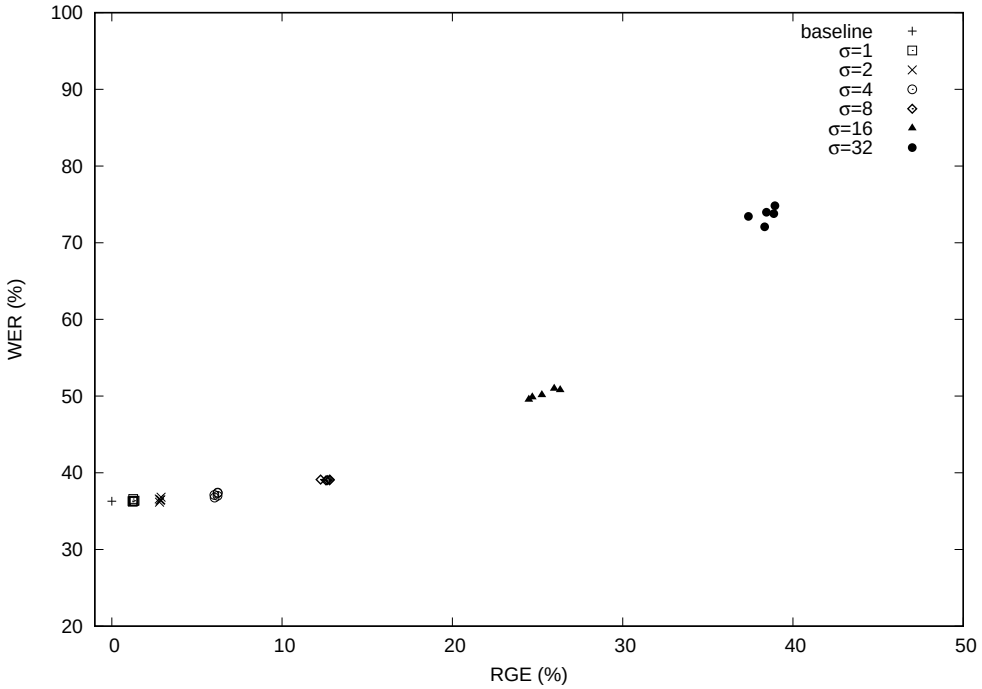


Figure 7.3: Figure presents the scatter plots of the different WER and RGE values obtained for the five different randomly generated baseline sets by each of the Gaussian distributions of different standard deviations (σ).

7.4.3 Discussion

As we indicated in Sec. 6.2 we need evaluation measures that *adequately* reflect how it will impact those who use the products yielded by the system. Furthermore, a very desirable trait of evaluation measures is that a relation or proportionality should be visible between the evaluation measure used to review the quality of the results yielded and the evaluation measure of the systems that use those results to perform a task. In our case the better the quality of the text line extraction the better the results of the handwritten text recognition system that uses the extracted text lines.

This last section has served a dual purpose. First, we reviewed how the competition measure, that was widely in use over a period of time, relates to the **WER** used to measure the quality of transcriptions in handwritten text recognition tasks. As we have seen in the experimentation performed on a real corpus, the competition measure might not be adequate to quantify the impact that the text lines will have on the system that uses them. Large accuracy improvements as per this error measure do not result in any significant gain on the systems that use them. Furthermore, there are cases that a gain as per the quality of the extraction measure actually ends up creating a negative impact on the handwritten text recognition system.

Second, we tested our graphical error evaluation measure **RGE**. As we saw in Sec. 7.3 baseline detection resolves most of the complexity of the text line segmentation problem, thus it seems

optimal to have an error measure that is based around the detected baselines accuracy. As we have seen in our experimentation the **RGE** is able to capture adequately the deviation between the groundtruth baselines and some synthetically produced baselines.

Not only is the **RGE** able to capture the error in a *meaningful* way it also reflects *adequately* how the quality of the baselines impact the systems that uses them. As we saw through out the experimentation performed on a real historical handwritten text, as the quality of the baselines deteriorated so did the transcriptions yielded by the HTR system. Furthermore a clear relation between the **RGE** and the **WER** could be seen.

Due to the above, we consider our evaluation measure based on baselines to be the most adequate way to actually measure the quality of a text line detection system. It is important to note, that the community, eventually moved to measures based on detection of the text lines [5, 12]. This change of focus on the evaluation measure happened during the course of the research performed for this thesis.

7.5 Statistical Text Line Detection and Classification

As we discussed in Chapter 3 the Text Line Segmentation task can be divided into the detection and extraction subtasks. In Sec. 7.3 we reviewed what was the actual importance of the detection phase in comparison to the extraction phase. As the results indicated, the resolution of the detection problem solves most of the complexity of the actual Text Line Segmentation task. Although the extraction has its importance, it is minimal and largely depends on the quality of the detection performed.

Due to the aforementioned importance of the detection phase, we focused our research on this task. Any improvement performed on this task will significantly boost all the existing text line segmentation techniques. As we saw in Chapter 2 most of the existing methods make heavy use of heuristics for the detection phase. Our intention, defined in Chapter 3, is to move away from these heuristic techniques and apply a stochastic framework to solve this task.

Additionally to our differentiated focus on detection we also intend to tackle the task of classification. This is due to our asseveration that trying to decouple classification of the elements in a page from the actual detection of the elements in a page is error prone.

In order to test the efficacy of our proposed approach for detection and classification we have performed a series of tests with different corpora where we measure the accuracy of our approach and compare it to a classical basic method. This classical method simulates the detection technique seen in a few published approaches, see Sec. 3.2 for details on it. The performance will be measured, due to the conclusions reached in Sec. 7.4, via the evaluation measures defined by us.

Additionally we plan to review the ease with which our framework can adapt to different corpus that require types of regions (text lines in this case) to be detected and classified. On top of this

we will experiment with the impact vertical layout models have on the system and how easily they allow us to incorporate expert knowledge.

7.5.1 Experimental Set-up

The series of experiments were performed on the CHRIS corpus described in Sec. 5.5 and on the WED corpus detailed in Sec. 5.6. As mentioned in chp. 5 these corpora were selected due to the fact that they both had the groundtruth region labels for each of the text lines and the coordinates of the baselines.

On one hand the CHRIS corpus is an adequate first test subject for our technology. As it is a prose text written by a single writer in one column, it allows us to review how our approach will fare with this simpler scenario. On the other hand the WED is a typical example of record-structured document that should greatly benefit from our machine learning based approach.

As we described in Sec. 3.3.3 we will adapt the VLMs according to the region elements (in this case line elements), and expert knowledge that can be incorporated to it. In this experiment we first consider two classical pure probabilistic models, which we named *1-gram* and *2-gram*. The unseen or underestimated transition probabilities of these models were smoothed via the the Witten Bell discount [1].

The *1-gram* model will be basically modelling the prior probability of each line region type while the *2-gram* model will be able to model the probability of the next region type considering the previous one.

In addition to the above, we wanted to illustrate the possibility of (deterministically) implementing known regularities of the vertical arrangement of text. To do so, we considered a simple line-number constraining finite-state model (later referred to as *M*axLines). Such a VLM only accounts for page images which contain a given maximum number of LRs. This could be helpful for a corpus where it is known that there is a set number of text lines, that may vary a little, per page. An example of this type of model can be seen in Fig. 7.4.

Using the training and development sets of each corpus we trained the STLCD models and tuned its meta-parameters: feature vector dimension D, HMM topology (number of states and Gaussians), number of Baum-Welch iterations, *VLM scale factor* and *LR insertion penalty* (see Sec.3.3). Meta-parameters were tuned to obtain the lowest development-set **LER** for each of the aforementioned VLMs : *1-gram*, *2-gram* and *MaxLines*.

After retraining with both the training and development sets, we run a single experiment on the test partition of each corpus using the optimized meta-parameters for each VLM.

For comparison purposes, we also applied the classic line detection approach outlined in Sec. 3.2, which only yields baseline positions, without any classification information.

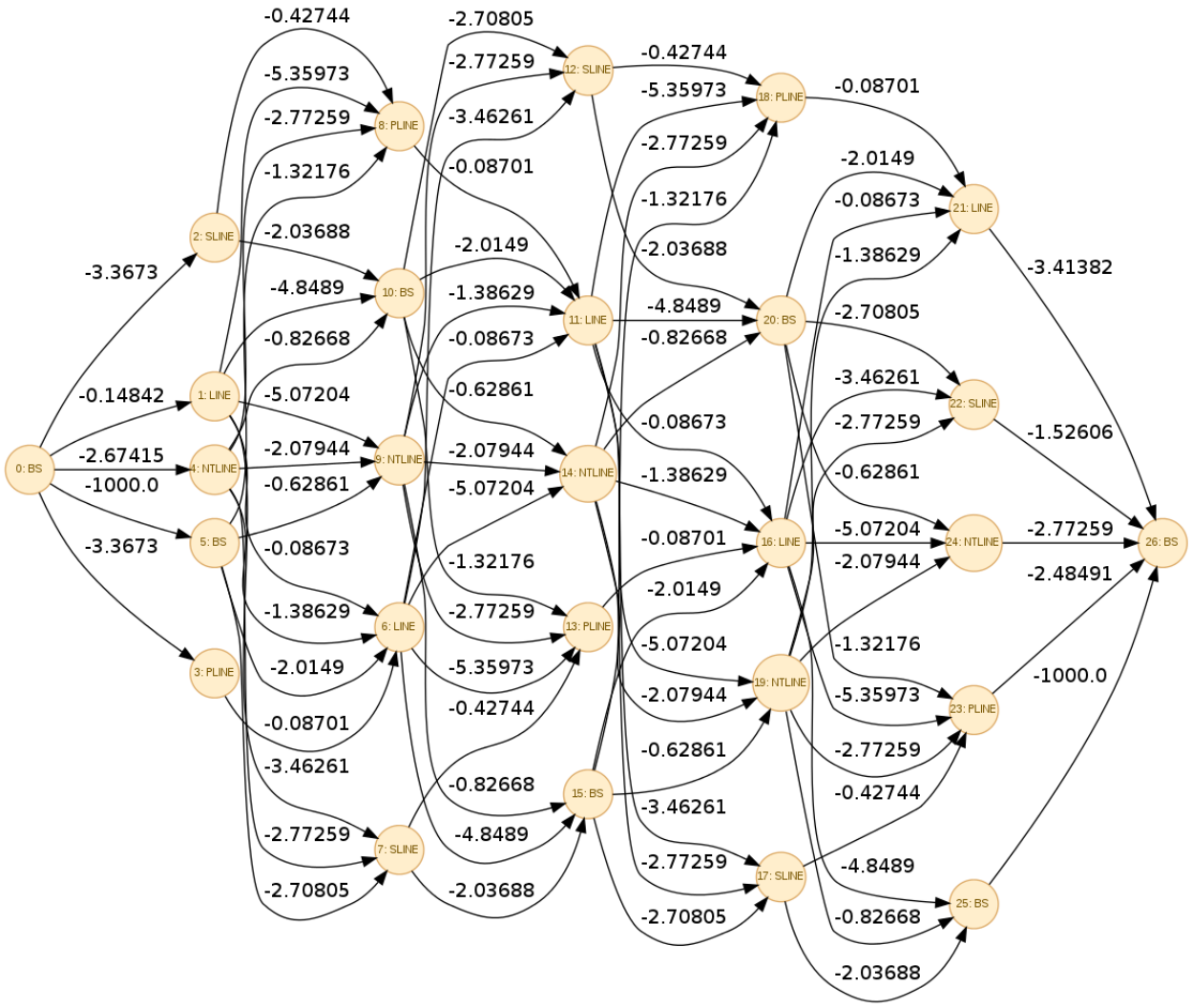


Figure 7.4: Example line-number constrained model where the number of text line regions has been limited to 5.

Vertical Position Coordinate Bias

An important advantage of the technique we described in Chapter 3 is that it does not need any information about the actual positions, heights, or boundaries of the text regions in the handwritten text image in order to train the system. This advantage is very convenient in practice as we will see in the different production practical scenarios in Sec. 7.6.

However, since the system lacks this position information during the training phase we are, in a way, allowing it to learn on its own what is the "optimal" region boundary concept. What the system learns may systematically differ from the *boundary* we have in mind.

If the corpus being worked on has a relatively small validation set with manually annotated

boundaries, as is usually the case, we can correct this systematic difference. By reviewing the difference between the calculated hypothesis boundaries and the groundtruth in the validation set this bias, which we will referred to as ξ, can be straightforwardly estimated.

Afterwards, in the test phase, ξ can be trivially used to fine adjust the baseline positions $\mathbf{b}$, provided by the system.

For this set of experiments we will analyse the bias of the system's baselines hypotheses with respect to the actual ground truth baselines of the development set. By doing so we will calculate the optimal bias correction term, ξ, with which we will improve the final **RGE** obtained for the test partition.

7.5.2 Results: The impact of the prior and bias correction

First, we report the RER and RGE achieved for the CHRIS corpus in table 7.4. Where the RGE mean and standard deviation are given as percentages of the text line average height (80 pixels).

Table 7.4: Detection, classification and segmentation errors (D-RER, C-RER and RGE) and bias correction (ξ), obtained on the CS corpus, using the classical approach and STLCD with different vertical layout models: *1-gram*, *2-gram* and *2-gram+MaxLines*. All the results are percentages, SER with respect to the average line width (80 pixels).

Approach	VLM	Validation				Test		
		D-RER	C-RER	RGE (st-dev)	ξ	D-RER	C-RER	RGE (st-dev)
STLCD	*1-gram*	1.10	6.34	12.4 (29.5)	2.5	0.86	10.10	9.7 (22.3)
	2-gram	0.90	4.85	12.1 (24.8)	0.0	0.70	8.34	11.5 (22.7)
	+MaxLines	0.00	4.76	12.0 (21.8)	-1.3	0.00	5.56	8.5 (19.8)
+Derivatives	*1-gram*	0.79	6.07	11.5 (11.6)	3.8	0.52	6.43	8.0 (8.1)
	2-gram	0.59	4.76	11.8 (11.4)	3.8	0.37	5.39	7.4 (6.1)
	+MaxLines	0.00	4.50	11.5 (10.0)	3.8	0.00	4.87	6.2 (6.3)
Classical Approach	–	9.53	NA	37.9 (26.3)	-31.3	12.18	NA	11.7 (22.9)

As shown in Table 7.4, the line detection error rate (D-RER) is below or around 1% for all the STLCD variants. For the *MaxLines* VLM, this error is null, which is to be expected. The line type classification error rate (C-RER) is of course always higher than C-RER. We observe that the results improve as more layout constraints and information are included into the VLM. For all the VLMS, *1*-gram, *2*-gram and *MaxLines*, the classification accuracy seems adequate. Which would allow us to use the obtained line type labels to improve the prediction capabilities of language models in handwritten text recognition tasks. It is important to note how easy and seamlessly we are able to add knowledge into the VLMs and the incredible impact that Prior information has on the accuracy of the approach. The above result is important due to two important reasons. Firstly, classical TLD techniques lack line classification capabilities. To our knowledge, our work is the first one that jointly provides the line-type classification and detection coordinates in a successful manner. Secondly, current TLD techniques being explored are finding it incredible difficult to include prior information into the detection process.

The inclusion of the HPP derivatives into the feature vector, described in Sec. 3.3.1, consistently reduces all errors types. Furthermore since derivatives provide a more explicit information about baseline position, this allows the EM algorithm to learn a more consistent "concept" of what a baseline is. The net effect of this, is that the RGE variance becomes much smaller and the segmentation bias remains stable, even when other experimental conditions change.

The proposed approach generalizes well. Test-set results are generally similar to the corresponding best validation results obtained after meta-parameter tuning. As expected, the RGE improves significantly in the final test-set, due to the fact that the models are finally trained with more training samples (both training and validation sets). It is important to note, that the biggest impact is thanks to the application of the bias segmentation correction term, ξ, estimated on the validation set. In comparison, results achieved by the classical approach are clearly worse: D-RER results are about ten times higher than the equivalent STLCD result. Furthermore, since it can not provide line classification results, we can not perform a comparison regarding this aspect.

In addition, the segmentation error (RGE) is significantly higher than that all of the STLCD variants, both for the validation and test sets. It also exhibits a much higher variance and bias. The large bias explains the great effectiveness that the bias correction has in this case. Our studied bias correction technique leads to a more than three-fold reduction of test-set RGE.

Next we will analyse the results for the WED corpus shown in Table 7.5. The observations and remarks that can be made about these results will be similar to the ones made for the CHRIS corpus. It must be noted that the WED corpus is more complex than CHRIS. This is due to the larger variety of line types and the smaller average line height (60 pixels). As a consequence, the relative benefits of using the different VLMs and the HPP derivatives are accentuated in this corpus. Due to the fairly regular register-based layout nature of WED pages, using more informed VLMs has a major positive impact in line type classification and detection accuracy. As described in Sec. 3.3.3 adding this type of prior information to our approach is fairly straightforward.

In this corpus, the inclusion of prior probability via the VLMs is particularly effective. Errors are (much) more than halved when using the *2-gram* or *MaxLines* VLMs with respect to using the least restrictive *1-gram* VLM.

The same is true for the use of the HPP derivatives, with more than 50% relative error reduction for all the VLMs. This reduction was smaller in the CHRIS corpus because of its larger inter line space, which made text line segmentation easier.

Table 7.5: Detection, classification and segmentation errors (D-RER, C-RER and RGE) and bias correction (ξ), obtained on the WED corpus, using the classical approach and STLCD with different vertical layout models: *1-gram*, *2-gram* and *2-gram+MaxLines*. All the results are percentages, RGE with respect to the average line width (60 pixels).

Approach	VLM	Validation				Test		
		D-LER	C-LER	SER (st-dev)	ξ	D-LER	C-LER	SER (st-dev)
STLCD	*1-gram*	4.31	13.93	14.8 (40.2)	0.0	3.61	13.18	13.2 (32.2)
	2-gram	1.84	6.13	11.6 (37.4)	-6.7	0.92	5.09	9.0 (24.1)
	+MaxLines	0.00	4.09	10.9 (33.1)	-6.7	0.00	4.83	9.0 (21.3)
+Derivatives	*1-gram*	2.41	8.82	10.2 (35.2)	0.0	1.32	5.28	9.5 (26.4)
	2-gram	0.58	2.81	9.7 (30.1)	0.0	0.31	2.46	9.2 (20.1)
	+MaxLines	0.00	1.90	8.5 (25.1)	0.0	0.00	1.30	8.1 (18.1)
Classical Approach	–	6.02	NA	26.0 (47.5)	-33.3	6.71	NA	12.3 (31.4)

7.5.3 Discussion

In this section we have shown the effectiveness of our approach for region detection and classification applied to text lines in practical scenarios. Our method described in Sec. 3.3 follows the same statistical framework that has been successfully adopted in many natural language processing tasks.

A new way of addressing text line analysis and detection has been proposed and assessed. The method makes good use of the sound statistical framework that has been successfully applied in many natural language processing tasks. Our empirical tests show that it provides very good results both in type classification and detection of text lines.

Our system is able to outperform a classical heuristic based line detection technique. This technique can be found in use in many of the complete text line segmentation approaches that were published during the research performed for this thesis.

It is very important to note, that our approach is able to incorporate prior information in an easy and effective manner. As seen in our results this has impacted positively the performance. Current state of the art approaches, based on recurrent neural networks, have no easy way to add this type of information in an easy manner.

Since our proposed STLCD approach is statistically based, it needs training data. This may seem to be a drawback with respect to training-free traditional techniques. However, the amount of training data required by our method is relatively small and, moreover the required annotation is extremely simple: just a sequence of the line regions labels for each training page image. If no classification is required the prerequisite of training data is required to a mere text line count per training image.

The actual impact, that this training data prerequisite has, is something that must be assessed. This study is crucial in order to perform conclusions regarding the practicality of our approach.

7.6 Text Line Detection in Real Production Scenarios

The application of machine learning solutions to real world problems is oftenly not considered adequate. The reason for this opinion is based on the fact that these type of methods classically have a steep entry cost.

Our approach, as described in Sec. 3.3, is based on a well known stochastic framework. Due to this, one might initially consider that the aforementioned criticism applies to our method. If we review the actual detail required in the labelling of our training data and the initial results presented in Sec. 7.5, we could consider that these issues do not seem to apply to our system.

In order to thoroughly address these concerns, we have performed several studies regarding the applicability of our approach to real production scenarios. These studies implied the application of our production process (described in Sec. 3.4) for the actual text line detection in actual historical manuscripts. The yielded text lines were evaluated by human experts and used for the actual computer assisted transcription of the documents.

In this manner we intend to review the applicability of our method to accelerate the production of groundtruth quality text line baselines. Through these experiments, performed under the duress of real production needs, we intend to showcase the small entry cost, accuracy and scalability of our method.

Additionally, we will be measuring how our method impacts the user effort required to produce groundtruth quality baselines. This evaluation will also allow us to study how our defined automatic evaluation measures (described in Chp. 6) relate to actual user effort.

7.6.1 Plantas Longitudinal Study

Throughout the research performed for this thesis, the developed methods were included and used in different projects. The *tranScriptorium* project was one of such projects. The *tranScriptorium* project pushed the boundaries of handwritten text recognition and worked close with its partners in order to leverage this technology to produce fully transcribed manuscripts.

One of the manuscripts transcribed in this project was the PLANTAS corpus described in Sec. 5.7. Inside the context of the project we tackled the first volume which is composed of approximately 1,000 pages (20,000 handwritten text lines).

The technology presented in this thesis played a crucial role in this large transcription endeavour. Our Statistical Text Region Detection and Classification method (see Sec. 3.3) was used to detect the text lines. While the extraction was performed using the distance map based dynamic programming algorithm (see Chapter 4).

Our text line detection approach was applied using the semi-iterative process described process described in Sec. 3.4. The study was initially performed in a more thorough manner on the PROLOGO chapter and was later extended to the rest of the volume.

It is important to note that the production transcription process was performed under the principles of a scientific study. The users of the different technologies were followed over the course of four months. We monitored their interactions with the different systems over time. To our knowledge, this can be considered the first evaluation of its kind in document layout and handwritten text recognition literature.

Set-up

The goal for the text line detection system was to yield groundtruth quality baselines for the handwritten text recognition system to have adequate training and decoding data.

The production process started with no actual data on the corpus. A single line count of the first page was used as the starting seed to train this model. Due to not having any means to fine tune parameters nor train the language models specifically for this corpus we used a set of standard values.

Essentially, the same feature extraction and HMM meta-parameters determined in other TLAD experiments with different handwritten data sets were adopted here. Additionally, they were not tuned throughout the work reported in this paper. In particular, the following meta-parameters were used: feature vectors of 7 dimensions, three 4-states Gaussian-density HMMs (one for each of the line shape labels: BS, IL, NL), grammar scale factor of 4, and a word insertion penalty of -128.

The VLM was fixed on the base of prior knowledge of the general structure of PLANTAS pages. Showcasing one of the great features of our method.

After this initial training with a single page, the iterative process was followed on the remaining blocks of the first chapter. For each block being processed, the text lines were detected automatically, revised by the user and added to the training set. After each iteration, the HMMs models were re-trained.

Afterwards, the $\sim 1\,000$ pages of the first volume of PLANTAS were processed, batch by batch, in the same manner.

Results

First we will evaluate the progress of our method throughout the different blocks of the PROLOGO chapter. Fig. 7.5 (top) shows the evolution of region detection error (RER), in this case the regions being the individual text lines. We measure the performance for each successive block of pages.

RER starts at 5% with just a single training page and it improves progressively as more pages are added to the training set, reaching at some points 0%. We can observe the same tendency for the RGE and URT in Fig. 7.5 (bottom).

We note that the task of line detection, basically determining the number of text line present in the page, proves to be a much easier task than obtaining its actual vertical location. Thus, the method requires a larger number of training to reach reasonable RGE levels.

Some small variances in the error plots shown in Fig. 7.5 can be observed. These small variances are due to the differences in the layout present in each block that the model has yet to learn. Regardless of the above, a clear correlation can be seen between the RER/RGE and the URT measure. The system yields better hypotheses as more data is made available for training, which in its turn effectively reduces the time needed by the users to review and amend the yielded hypothesis. The above correlation between our measures and the user review time supports their adequacy (see Sec. 6.2).

If we look at the first result, achieved with just the initial training seed of a single page, it can be basically considered equivalent to the effort required to manually detect and locate the baselines. This allows us to see the time reduction our system provides over manual ground-truth production.

Fig. 7.6 provides additional detail regarding how the graphical error (vertical difference between the hypothesis and the final user reviewed baseline) evolves during the iterative process. In each histogram we show the distribution of this graphical error at different time windows.

Initially, the graphical error tends to be large (about 50% of the average line height). After training with just 10 pages, the relative errors become smaller, but are very disperse and quite large in some cases. This improves further after the iteration where 20 pages are used for training. Finally, after training with 30 pages, the errors are concentrated around small values.

We can explain this variations over time in an intuitive manner. Initially, in the first batches, the HMMS do not have enough training data to be able to pinpoint the correct baseline position. As more training data is made available, the system starts to provide hypothesis that are closer to what the user expects. This reduction in graphical error goes hand in hand with the reduction of user review time required to correct them. We can clearly see the evolution of the baseline positional error in the two examples presented in Fig. 7.7.

After the processing of the initial PROLOGO chapter the work was extended to the rest of the volume. For the rest of the volume we no longer measured the RER (practically 0 at the end of the first chapter) nor the RGE. The performance for the rest of the batches of the volume were evaluated by means of the user review time.

Fig. 7.8 shows the evolution of the review time for the subsequent batches. As happened in our more detailed study inside the PROLOGO chapter, the effort required to review the batches was reduced over time. If we observe the cost of reviewing the first block of the PROLOGO chapter ($\sim$ 350 seconds) and compare it to time taken for the last batch ($\sim$ 50 seconds) this implies an 85% reduction in user review time.

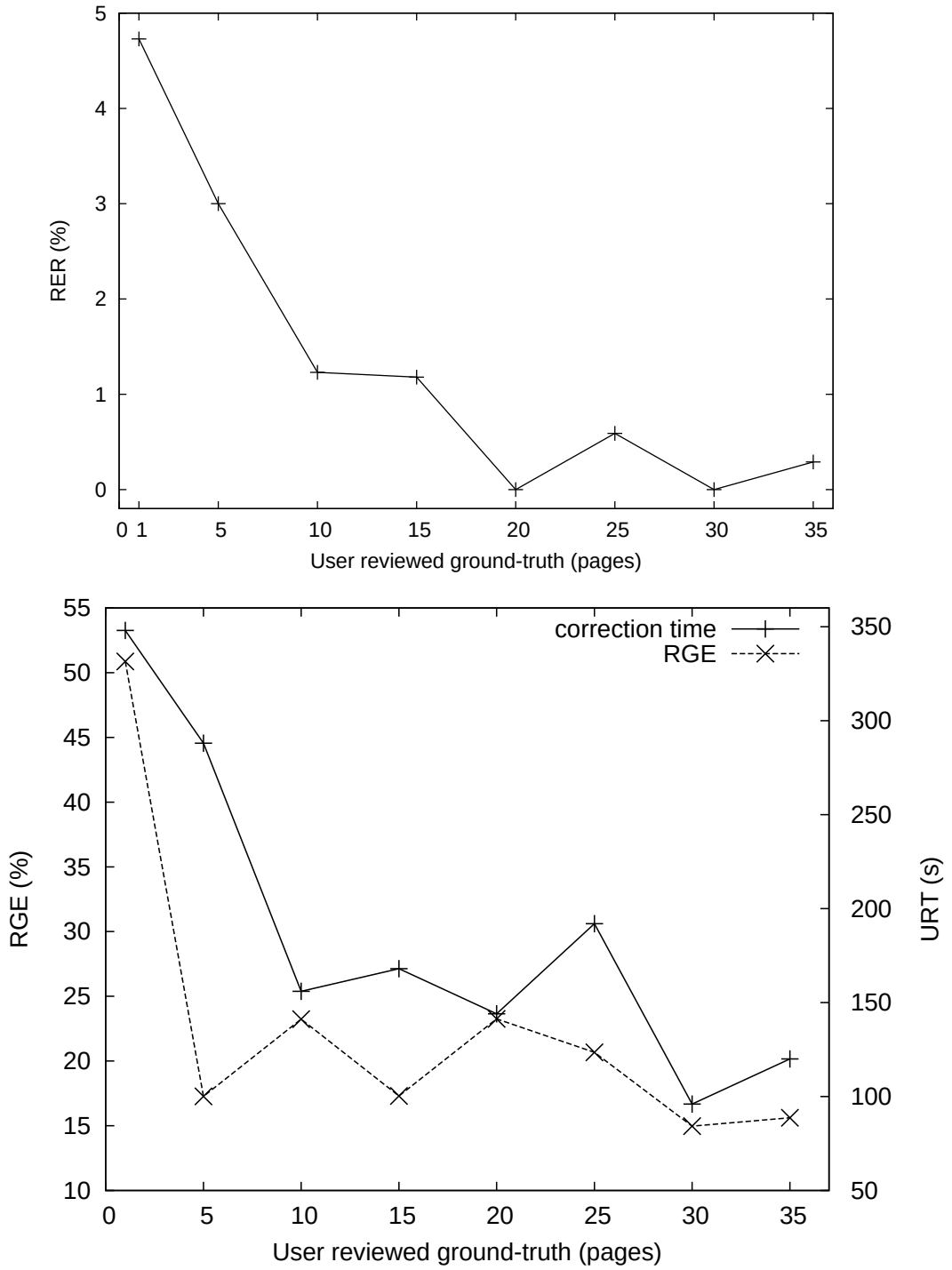


Figure 7.5: Evolution of the performance of the proposed semiautomatic TRDC process with the number of user-revised pages: Top: Region Error Rate (RER) applied to lines. Bottom: Relative Geometric Error (RGE) and User Revision Time (URT). RGE with respect to the average line width (80 pixels).

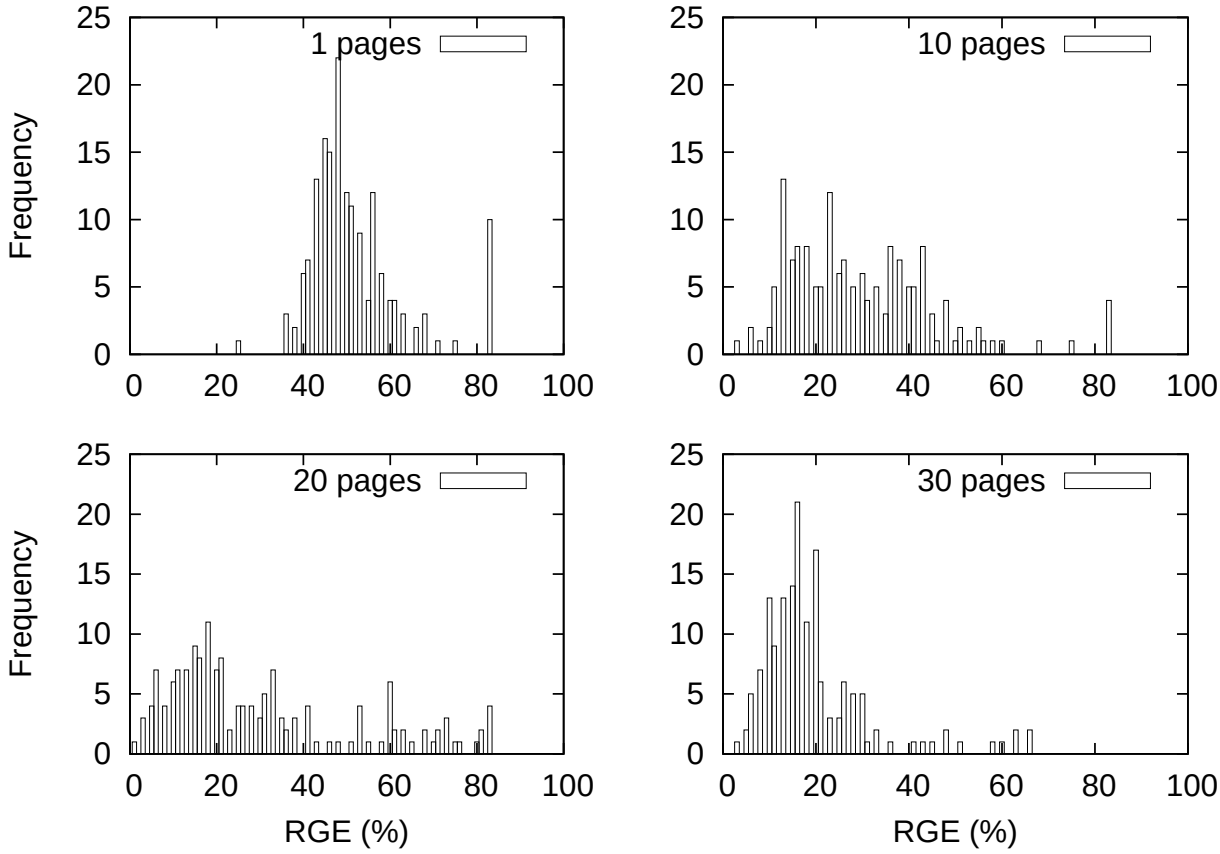


Figure 7.6: Histograms for the RGE (in %) of the hypothesis yielded by the system trained with different number of user revised pages (1,10,20,30).RGE with respect to the average line width (80 pixels).

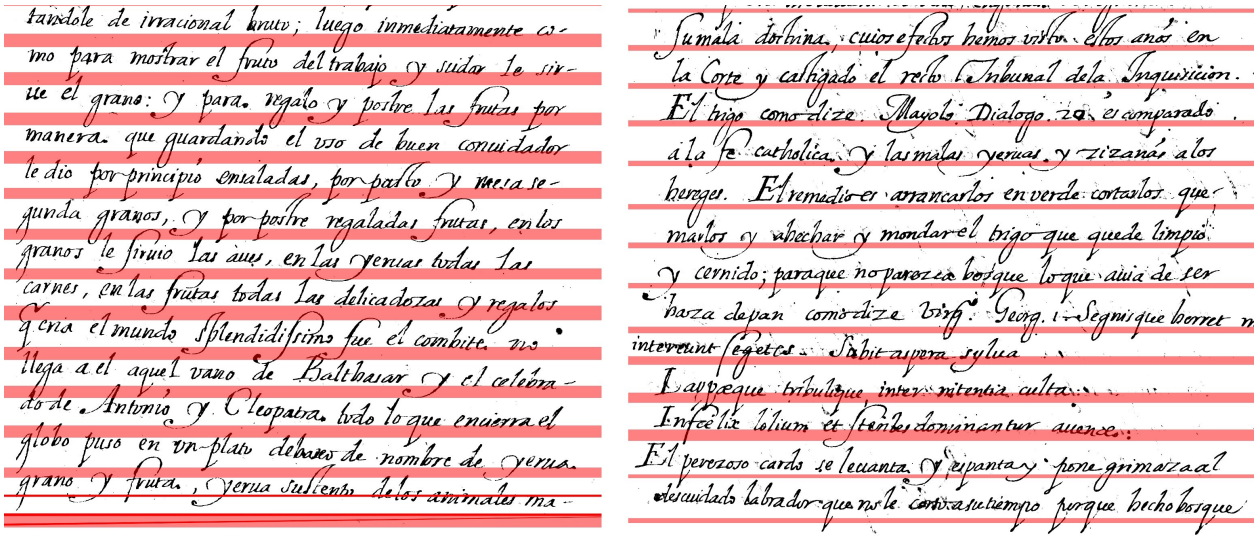


Figure 7.7: Sample pages showing visually (in red), the difference between the system baseline hypotheses and the corresponding user-corrected result. The left side sample (RER: 6%, RGE: 50%) corresponds to just a single training page while the right side (RER: 0%, RGE: 22%) was obtained after 30 user revised pages.

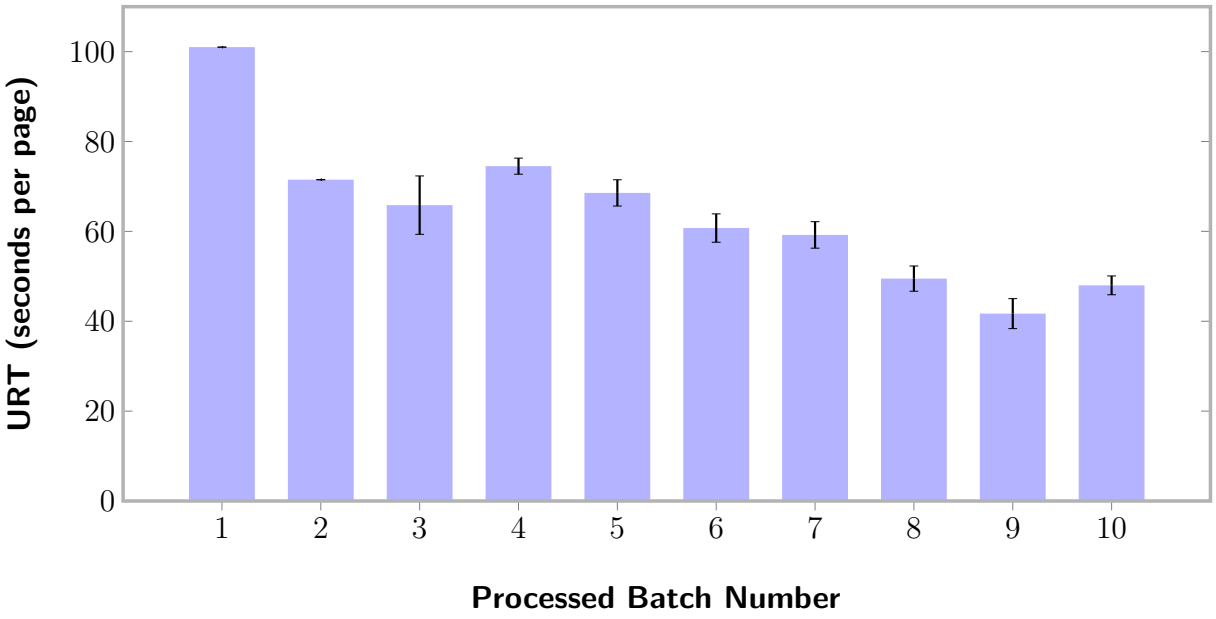


Figure 7.8: Computer assisted line segmentation: evolution of user review time (URT) with the successive batches of page images processed. Thanks to incremental model retraining, supertranscriber's reviewing effort significantly decreased over time.

Longitudinal Study Discussion

As we have seen during this large longitudinal study, our probabilistic approach can be applied adequately to text line detection in a production scenario. Not only does it provide good accuracy, comparable to or better than that of traditional heuristic approaches, it also has a very small entry cost in terms of the user effort required. The method is able to yield adequate results from the very beginning with just a line count of a small set of training pages. Due to the above commented qualities we can conclude that his approach is a very good choice for real life, practical usage.

As per the results obtained in the study the approach seems to be both robust and scalable. The method provides more accurate hypotheses as more labelled data is available for training and this effectively reduces the required user revision time.

This production scenario has also allowed us to showcase the ease with which our graphical region detection and classification approach can be adapted to the subtask of line detection.

7.6.2 RSEAPV Study

Our technology was also used in the full transcription of the RSEAPV corpus (described in Sec. 5.8). Our Statistical Text Region Detection and Classification approach was used together with the interactive HTR work-flow known as CATTI [14] to form an end-to-end system.

Set-up

In this scenario the STRDC framework was used to perform the text line detection. The baselines yielded were used to extract the text line images with which the CATTI system would be trained. In this case a single experiment was performed by training the system with 22 pages (just a line count of each page was required) in order to evaluate if the technology could be applied automatically to the rest of the manuscript and possibly the whole RSEAPV collection.

Results

Table 7.8 shows the results obtained in the text line segmentation process. We obtained a RER of 2.6%, which means that, for every 100 lines, less than 3 caused issues to the line detection system. Furthermore we provide the RGE measure that indicates the geometrical accuracy of the detected horizontal baseline coordinates with respect to the corresponding ground-truth baselines, in average the detected lines are close to the actual baseline reference and as we will see later are of an adequate quality for the HTR system.

Table 7.6: RER and RGE results of the text line segmentation process.RGE with respect to the average line width (90 pixels).

RER (%)	RGE (%)	
	Average	Standard deviation
2.6	11	30

An important aspect to remark in these experiments is the really small number of pages (with no segmentation information used) required to obtain good results. Only 22 pages were necessary to obtain a detection error below the 2.6% mark. The text lines extracted with our method resulted in adequate HTR results: 35.9% WER and 16.4% CER results.

RSEAPV Discussion

In this case our technology was used in a small feasibility study. The method was able to provide automatically adequate text line detection and localization to be used by an HTR system. This result adds to the notion that our approach is able to provide usable results with a very small entry cost.

7.6.3 Automatic Text Alignment

The methods described in this thesis have been used in the different projects in which we have participated. Usually, our methods are used for region or line detection and extraction as part of a transcription end-to-end system or an information indexing system. Regardless of this typical usage, our technology can also be used to automatically detect regions for other purposes.

In this concrete case, the technology was used in order to automatically align handwritten images with priorly performed transcriptions for which there was no alignment information. Specifically it was used in the HATTEM corpus (see Sec. 5.4) in order to automatically create more groundtruth data that could be used to train and enhance the HTR results.

The lines detected by our approach were transcribed by an HTR system in an automatic manner. This newly transcribed text was compared to the existing human transcribed text for that page via the Levenshtein distance. Depending on the result of the comparison the new pair of text line image and transcription was added to the training corpus in order to improve the model. Information regarding specifics of the automatic alignment technique and confidence measure can be read in [15].

Set-up

We carried out a series of experiments in order to assess the capability of an automatic alignment method based on our automatic statistical text region detection and classification approach. This will be performed by evaluating the impact that the new pairs of text and line images introduced into the training corpus has on the performance of the HTR system.

In an indirect manner, the performance gains obtained due to the new corpus training data provided by the alignment system, validates the usefulness of our region detection technology.

For this experiment, we used the partitions described for the Hattem corpus (see Sec. 5.4) to perform cross validation experiments. For each of the results we will present later 8 rounds of experimentation were carried out. In each round 7 partitions were used for training leaving the 8th for the test evaluation.

With this initially trained system we used the developed automatic alignment tool on a set of 263 pages of the same corpus. These pages were automatically segmented and transcribed with the initial HTR system.

With respect to the text line segmentation, the same feature extraction and HMM meta parameters determined in previous experiments with different handwritten data sets were adopted here.

The pairs of aligned text and text line images that surpassed a threshold were then selected and used to train our final HTR system. This final HTR system was also evaluated against the unused test block.

Results

Table 7.7 shows the results obtained with the three different training sets. The first row are the results for what can be considered the baseline system (B-HTR). This HTR system has only been trained with the training samples of the 7 training block taken out of the GT. The second row (LM-HTR) corresponds to an HTR system that uses the same optical models trained in B-HTR but has been enhanced with a better language model. Basically this enhanced language model was created with all the additional non-aligned transcripts of the extra 263 pages. The final row (ALIGN-HTR) contains the results of the HTR system that not only uses the enhanced LM but has also used the automatically aligned lines (in addition to the GT lines) to train the optical models.

Table 7.7: WER and CER results obtained by the different trained HTR systems.

	WER	CER
B-HTR	33.1	17.6
LM-HTR	26.1	12.2
ALIGN-HTR	24.2	11.1

From Table 7.7 we can conclude that the automatically added training data had a positive impact on the HTR system. Although the results were expected to be better, we managed to obtain positive results with no type of tuning of our framework. Additionally there was no human involvement required for this performance gain.

Alignment Discussion

In this specific scenario our technology was used as part of a fully automatic system. The end-to-end system was developed in order to increase, in a non-supervised manner, the amount of training data. Our approach was able to provide adequate text line detection with which an alignment between the already existing transcription and the newly extracted text line image was performed. Although better results were expected our method was able to provide reasonable text line region hypothesis with no type of actual tuning.

7.6.4 Discussion

Throughout this section we have seen our developed method being used in several real life production scenarios. These scenarios have covered the two main ways of using the framework: semi-automatic or completely automatic.

In the semi-automatic process we proved how our process can reduce the user effort required to produce ground-truth levels of quality in baseline detection. In the automatic process we showcased how our process is able to produce adequate results that have a positive impact on the system or process that depends on the yielded results.

In both ways of using the developed technology, we have proven that it has a very small entry cost in terms of the user effort required. When no classification of regions is required, as happens in simple line detection, only a line count of each of the training pages is required to train the system.

Furthermore, the method is able to provide adequate results without any actual fine tunning of the training and de-codification parameters. Additionally, as proven in the longitudinal study, the system is able to escalate adequately since it provides better hypothesis as more training data is provided.

Through the results obtained in the different studies we believe that we have proven that our machine learning based method is applicable to actual practical production scenarios.

7.7 Statistical Region Classification

As described in Sec. 3.3 our method is designed to detect and classify the different vertical regions that compose a page. The framework will adapt to the desired type of vertical regions depending on the *vocabulary* of regions the user defines. These vertical regions can be coarse like paragraphs, diagrams, tables, footnotes, etc. or much finer reaching like text lines.

In the past experiments, presented in this chapter, the application of our method has been more focused on the text line scenario. In truth, the designed region *vocabulary* contained a mixed set of coarse and fine regions. We used the flexibility of our system to tackle in an easy manner the issues created by large regions when trying to detect small regions like text lines. For example, the large blank spaces that can be found in all corpus or the large diagrams like the ones found in the PLANTAS corpus.

In any case, in this section we will showcase through an specific experiment the pure large region detection capability of our approach. Additionally, the importance of region classification and the potential impact it can have on other systems will be made clear. We also study the impact that over-classification has on the system's performance. Furthermore, we will do all of the above while tackling a completely different type of document.

7.7.1 Experimental Set-up

This experiment was performed with several objectives in mind. First, we wanted to tackle a more pure text region problem. Secondly, it would be interest to apply the framework to a radically different type of document that showcases the positive impact of not just detecting but classifying the regions.

Due to the aforementioned objectives we chose the CAPITÁN corpus. This corpus composed of different musical scores allows us to demonstrate the flexibility of our approach. By just adapting the vertical region vocabulary to the list of regions described in Sec. 5.9 we can tackle region classification in Music Sheets. This adaptability is only possible since our approach is designed to tackle region classification. Without the notion of *region classes* at the core of the approach it would require large modification to apply our method to such different type of documents.

Due to the large similarity between the pages of the different musical scores in the corpus, we were able to straightforwardly define a VLM that could account for the variability in them. The model, shown in Fig. 7.9, deals adequately with the arbitrary number (or range) of expected pairs of staff-lyrics regions.

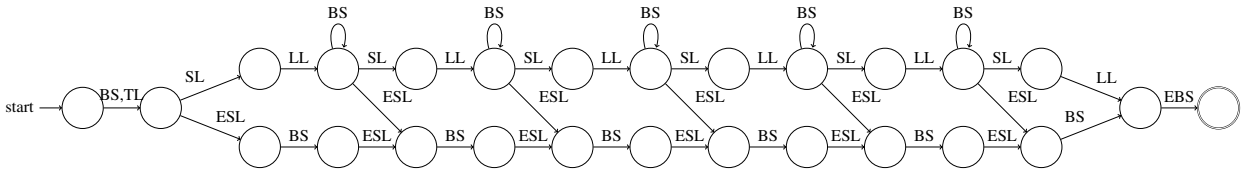


Figure 7.9: Deterministic finite-state automaton used as a vertical layout model (VLM) for CAPITÁN page images.

With respect to the set of parameters for feature extraction, training and decoding meta-parameters no fine tuning was performed. We used a set of standard values that have provided successful results for different handwritten data sets. Parameters were set as follows: feature vectors of 14 dimensions, 4-state HMMs (one HMM for each of the region classes described in Sec. 5.9) with 8 Gaussians per state.

We computed the RER and RGE for the different levels of detail used in the groundtruth labeling. In this experiment we studied five different complexity levels:

- *Detection of foreground regions*: being able to differentiate between foreground regions and the background.
- *Staff and Lyric differentiation*: the basic complexity to actually attack region detection in music sheets. This complexity level included the five main region types: title line (TL), blank Space (BS) , staff lines (SL), empty staff lines (ESL), and lyric lines (LL).

- *Multiple staff sub-classes*: adding further complexity by allowing the staff class to be divided into sub-classes. Taking into account if the staff region contains: ascending notes, descending notes or both.
- *Multiple lyrics sub-classes*: dividing the lyric lines class into normal lyric lines or short lyric lines depending on their length.
- *All sub-classes*: using all the sub-classes of the main staffs and lyrics classes.

These different will allow us to see and discuss, how this difference in complexity of the labeling impacts the results.

7.7.2 Results

Table 7.8 presents the detection and classification results obtained for the five different types of labelling defined previously. The average height of the different regions that compose a page, used for calculating the RGE, was 185 pixels.

Table 7.8: Region error rate (RER) and relative geometric error (RGE) obtained for various levels of region labelling detail.

Labeling detail level	RER (%)	RGE (%)	
		Average	Std. dev.
Foreground Detection	1.1	3.0	3.1
Staff / Lyrics	4.6	3.0	3.1
Multiple Lyrics Classes	6.9	3.2	3.5
Multiple Staff Classes	28.0	3.9	5.2
All sub-classes	30.3	8.5	11.2

The qualitative detection error (RER) is less than 5% for both foreground detection and the staff/lyrics classification.

Thus the system is not only able to separate the different regions of a music sheet but also differentiates between the most important region classes; i.e., staff and lyrics. Being able to reach this level of performance by just adapting the vocabulary to the required regions with no actual fine tuning is a major gain in comparison to any heuristic approach.

Being able to add prior knowledge to the system via the VLM plays a major factor in achieving these results. This feature of our approach is not easily reproducible when using other machine learning models.

As expected, as the number of sub classes of staff or lyrics regions increases, the classification error rises.

In the case of the multi staff classification result the increase in error is relatively large. This is clearly due to the small visual differences between SL, SL-A, SL-D and SL-AD regions. The small visual difference gets smaller when we analyse these staff elements together with the overlapping elements of adjacent lyrics regions. On the other hand, the small RER increment in multiple lyrics classification has been observed to be mainly due to confusions caused by noise issues.

The RER error increases as expected when using at the same time both the lyrics subclasses with the staff sub-classes.

The geometric baseline detection error was very low (less than 4% in most cases). The RGE evolved as expected; the increase in the labelling detail implied a reduction in the relative amount of training samples per class which in the end impacted the HMM model training. The negative impact in the HMMs reduced their precision which was adequately registered by an increase of the RGE.

We should point out, however, that this high segmentation accuracy can be further improved. In fact, we observed that the baseline positions yielded by the system tend to be slightly biased which could be solved if we applied the the bias segmentation correction term.

An important aspect to note is that in order to produce all the results presented only a small number of pages (with no segmentation information required for training) were required. With only 50 pages out of the thousands present in the CAPITÁN collection, we have an adequately trained system that can be used to process the rest of the pages.

7.7.3 Discussion

Via the above experiment we have demonstrated how our approach can be applied to a pure region segmentation and classification task.

We performed this experimentation on a music sheet corpus that can be considered radically different from the usual documents used in document layout experiments. We were able to perform this experimentation with just some small changes to the modelling configuration in order to adapt it to the new kind of regions.

As always, our system is able to use any type of prior information by incorporating it via the VLM and is able to provide adequate results without the need for specific parameter fine tuning.

In the case of automatic music score transcription, the staff and lyrics have to be transcribed by different systems for obvious reasons. This task showcases the need for region classification that our approach provides out of the box.

7.8 Enhancing Statistical Region Classification with Word Probabilistic Indexes

As we saw on the previous sections, we obtained very adequate results regarding detection and classification of regions. These results were achieved when we applied our approach to larger text regions like paragraphs, staff areas in a music score or diagrams or when applied to smaller regions like text lines.

All of the previous experiments relied on graphical features to do the detection and classification of the regions. Although this yielded adequate results, there are scenarios were using these features alone will not yield good performance. This is specially true for classification.

When the regions types to be classified do not present substantial graphical differences we must use other features to differentiate them. Features based on the contents of the text would help tremendously, unfortunately that usually required the layout to be completed and the text to be automatically transcribed. Fortunately, there are currently ways to extract probabilistic information of the words contained in a page that allows us to use this aspect for our classification without the need of performing a fully fledged transcription of the page.

For this section of the experimentation chapter we will focus on the application of word probabilistic indexes (see Sec. 3.6) to enhance our region classification approach.

7.8.1 Experimental Set-up

In order to test the enhancement capabilities that the word probabilistic indexes provide for our region classification approach, we need a classification task where the different regions do not present much graphical differences. The CHANCERY corpus (see Sec. 5.10) provides such an scenario.

The CHANCERY corpus is composed of different *Acts* where the different rulings of the kings regarding the organizational bodies that composed the kingdom were registered. Unfortunately these Acts, basically written paragraphs of text with specific wordings and formula, sometimes encompass more than one page. In order to make the information more searchable we need to identify when *Acts* start, finish or both start and finish in a page.

Unfortunately, there is no graphical information from which to discern:

- If a text paragraph is a continuation of an act that started in the previous page.
- If a text paragraph ending, actually represents the ending of the act or if it will be continued in the next page.

We must rely on the contents of the text in order to differentiate these cases. Since *Acts* start and end with very particular words, any information regarding the presence of such words (even if it probabilistic) will aid us significantly. In this case we will use both graphical and text content features to perform this classification.

We used the information present in the automatically calculated word probabilistic indexes to calculate for each pixel line the probability of existence of words for three different categories: beginning of an act, middle of an act and ending of an act. These categories corresponding to the layout elements considered for this corpus. Due to the use of specific wordings and formulas to start and end an act we can select a specific set of words for each act subregion that allows us to differentiate between them. We used the aid of an expert in order to provide a set of words that would aid us in differentiating each category. The vocabulary for each paragraph zone category, could have also been selected in an automatic manner.

For this specific corpus the feature vectors consisted of 9 dimensions. The page was divided into three columns, for each column we calculated the horizontal projection profile and its derivate (see Sec. 3.3.1) which resulted in 6 graphical features. The other 3 dimensions were calculated by summing the probabilities for each word to the category they pertained (see Sec. 3.6).

The training and validation partitions of the CHANCERY corpus were used to fine tune the parameters. The final training and decoding parameter values were the following: *training iterations* 10, *WIP* -512, *GSF* 32 with an specific number of states selected for each region {I: 5, M: 5, F 5, O: 2, C: 2, BP: 7, EP: 5} .The test partition was used a single time in order to obtain a final result with unseen data.

7.8.2 Results

In Table 7.9 we can see the different error results obtained for this corpus with the different features sets. As seen in other experiments the detection error rate is much lower than the classification error rate due to the increment in complexity of the task. In fact, simple detection of the different Acts, basically paragraph detection, is performed adequately with both types of feature vectors.

Table 7.9: Detection and Classification region error rate (RER) and relative geometric error (RGE) obtained with different types of features. Due to the large size of the regions being detected the average height to compute the RGE was artificially considered to be 100 pixels.

Features used	D-RER (%)	C-RER (%)	RGE (%)	
			Average	Std. dev.
Graphical features	6.82	77.93	6.8	3.9
Graphical + Textual features	7.04	28.81	10.2	3.1

In this case, due to the small (or non existent) graphical differences between the region types we are classifying, the classification performance of the system falls substantially when only using

our usual graphical features. When we add to the feature vectors the aforementioned textual based dimensions the systems Act classification performance improves dramatically.

Contrary to what could be initially thought, the system that only used graphical features shows a marginal improvement in D-RER and RGE evaluation measures. We believe this to be due to two causes. First, there are acts that end up with a signature section and some information on the date and place the existence of these words has caused some slight problems to the classifier that uses text features as it expects this information at the beginning of an act. Second, the bleed through present in the pages and the issues it creates to the process that automatically detects the text and calculates the word probabilistic indexes. An example of this issues created by bleed through can be seen in Fig. 7.10.

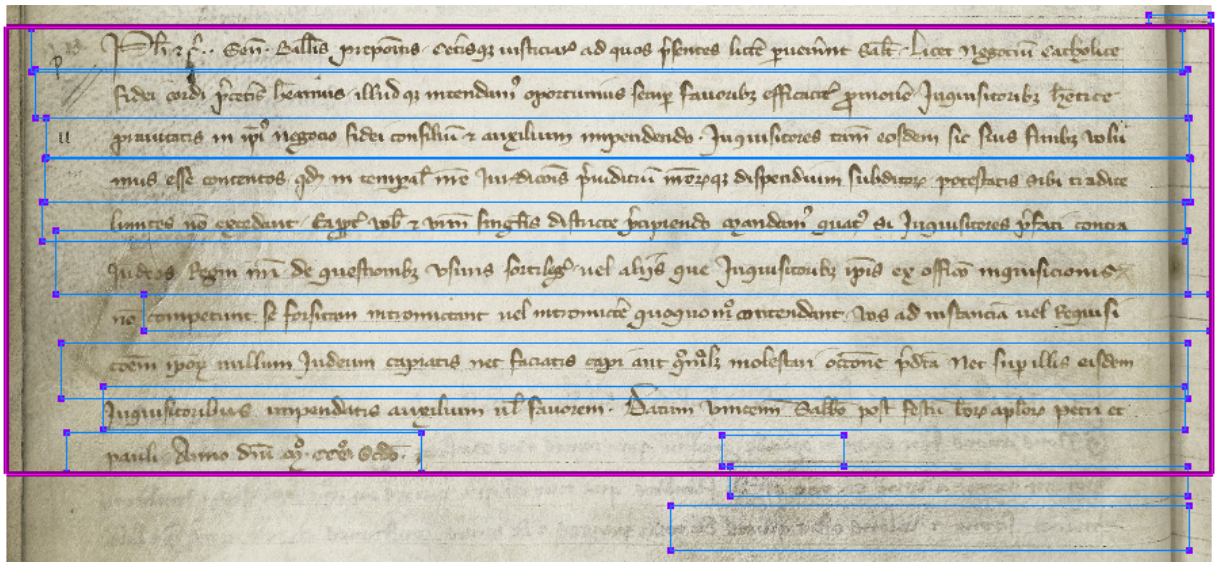


Figure 7.10: Sample page segment showing visually (in fuchsia), a Complete Act. Inside the Act we can see the automatically detected text zones (in light blue) used to calculate the word probabilistic indexes. In the bottom right corner, we can see examples of strong bleed-through falsely detected as foreground text that was included in the word probabilistic indexes calculation.

7.8.3 Discussion

In this section we have shown the positive impact that the information present in word probabilistic indexes can have on text region classification. Due to the current possibility of having such information, before text regions and lines are extracted or a full transcription is performed, document layout methods can improve their performance significantly.

This work is, to our knowledge, the first time features derived from word probabilistic indexes have been used in document layout methods. We firmly believe, that leverage the information provided by the word probabilistic indexes as textual content features will impact greatly the document layout analysis state of the art in the years to come.

7.9 Enhanced image preprocessing with Convolutional Neural Networks

As covered previously (see Sec. 3.5) there is a renewed interest in Deep Neural Networks (DNN) and applying them to solving different tasks. At the time the research work for this thesis started the Document Layout Analysis scene was dominated mostly by heuristic based methods. Usage of DNNs did not truly start to pick up until advances on the usage of Graphical Processing Units (GPUs) for matrix computation were made publicly available.

We can observe in the last major DLA related congresses several articles on the application of Convolutional Neural Networks (CNNs) to the HTS tasks. Although some articles do cover how to actually produce an actual region or baseline [8, 13, 16] from the results provided by the CNN most only provide as results the probability map calculated by the CNN [3, 4, 11, 18].

The main product of the CNN is a probability map regarding each of the trained page region classes. Although this probability map simplifies greatly the HTS task it is important to note that it the map itself does not actually represent a final segmentation result. The decision making in order to calculate the segmentation frontier or baseline is performed by different post-processing methods. In Fig. 7.11 we can observe the probability map computed by the CNN and compare it to the image preprocessed the classical approach described in Sec. 3.2.1.

Figure 7.11: Figure shows a side by side comparison of an original image (centre) with the classical preprocess resulting image (left) and the resulting image of displaying the per pixel text line probability computed by a CNN (right). Sample page extracted from the HATTEM corpus.

Due to all of the above we can actually consider the CNN as a preprocessing technique that is able to enhance the specific page regions we are looking for. This implies that we can use our Gaussian HMM based framework to make decisions using the CNN as a feature extractor. This NN plus HMM tandem approach has also been used in ASR [10] and HTR [2].

In this last section of the experimentation chapter we will focus on enhancing the results of our HMM detection model by using it in a tandem approach with CNN [13]. We will use the CNN yielded probability maps as features for our Gaussian HMMs.

7.9.1 Experimental Set-up

In order to test the enhancement capabilities of the CNN-HMM tandem approach, we need a task in which our classical preprocess scenario has issues and difficults the detection and classification of page regions by our HMM framework. Furthermore, we need a task that might prove to be difficult to an approach that performs a heuristic post-process of the probability maps yielded by the CNN. As we saw in Fig. 7.11 performing text line detection in the HATTEM corpus (see Sec. 5.4) proves to be an adequate scenario for this test.

A U-net CNN [13] was used in order to generate the text line foreground probability map. A general CNN model, trained with images of a long list of historical corpus, was used for this experiment.

For the HMM training we randomly selected 20 pages from the corpus and uses the other 20 pages to test the accuracy of our system. The VLM used will be similar to the one used in our *production scenarios* experiments (see Sec. 7.6) with a strong prior towards single column text line detection.

For this specific experiment the feature vectors consisted of 8 dimensions. The page was divided into 4 columns, for each column we calculated the horizontal projection profile and its derivative (see Sec. 3.3.1) which resulted in the aforementioned 8 graphical features.

The randomly selected training partition of HATTEM corpus was used to fine tune the parameters. The selected final training and decoding parameter values are as follows: *training iterations* 7, *WIP* -128, *GSF* 4 with 6 state HMMs. The test partition was used a single time in order to obtain a final result with unseen data.

In this experiment we will compare the following system set-ups:

- The **Classical Preprocess + HMM** system used mostly in this thesis with its HMMs and VLM trained as described above.
- A state of the art **CNN + Heuristic Post-process** system [13].

- A novel **Tandem System (CNN + HMM)** that utilizes the CNN to generate a text line foreground probability map and a HMM that uses the yielded probability map as an image on which it calculates the graphical features.
- Using the **Tandem System (CNN + HMM)** in combination with the information of a projection profile text column detector to clip the yielded baselines.

7.9.2 Results

In Table 7.10 we present the different error results obtained by the three different approaches detailed before. First of all we can already observe a massive improvement between both approaches that use HMMs. The use of CNNs to preprocess the image is a clear advantage over classical heuristic preprocessing. The CNN method + the heuristic post process pays a hefty price in this experiment and our evaluation measures due to the combination of two facts: our evaluation measures penalizes considerably missing or extra regions and the method has no easy way to add prior information. The impossibility to add prior information allows the CNN and the heuristic process to divide erroneously text lines.

Table 7.10: Result table with the Detection region error rate (D-RER) and relative geometric error (RGE) obtained with the different systems. Due to the large size of the regions being detected the average height to compute the RGE was artificially of 130 pixels.

System used	D-RER (%)	RGE (%)	
		Average	Std. dev.
Classical Preprocess + HMM	9.2	36.15	100
CNN + Heuristic Post-process	38.79	37.69	100
Tandem: CNN + HMM	4.1	16.57	27

Despite the results shown in Table 7.10 it is important to note that the actual base line detected by the CNN + Heuristic Post-process is more detailed than the HMM approaches. It is adjusted in a more intricate way to the textline and adjusts to the textline ends. In order to review this issue we perform an evaluation of all methods with the *TBES* measure in Table 7.11.

Table 7.11: TBES result table for the different systems used in this experiment. We additionally added an extra value by adjusting the text columns via a simple heuristic method to automatically clip the text lines.

System used	Precision	Recall	F-measure
Classical Preprocess + HMM	0.57	0.88	0.70
CNN + Heuristic Post-process	0.71	0.93	0.81
Tandem: CNN + HMM	0.56	0.93	0.70
Tandem: CNN + HMM (column detect)	0.86	0.90	0.88

Table 7.11 presents the TBES results for our three initial systems. As we can see via this evaluation measure the CNN + Heuristic Post-process seems to yield graphically better baselines. Although this method still has the issues regarding erroneously divided baselines it is important to note this fact. Although our initial tandem system scores lower in the precision measure due to the fact that our HMM framework yields straight baselines from edge to edge it still performs adequately. In fact if we detect the text columns automatically via a simple heuristic method we can use this info to clip the baselines and overcome this issue as we see in the fourth row of this table.

7.9.3 Discussion

In this section we have shown the positive impact CNNs have on the HTS task. Our simple tandem combination of CNNs and HMMs already showcases the superiority of the CNN direct processing of the original images has over classical image preprocessing.

Furthermore, the tandem proves the usefulness of our HMM based framework to perform the actual region detection. As said before the probability map yielded by the CNN is quite impressive but the model does not actual make any final decision regarding the regions nor does it allow us to easily include prior information.

7.10 Chapter Conclusions

In this chapter I have presented the different experiments performed in order to prove the different scientific outcomes we listed in Chapter 1. Our empirical studies reviewed how the methods defined in chapters 3 and 4 fared in different scenarios. The scenarios, defined by the corpus listed in Chapter 5, have allowed us to evaluate our methods and confirm our initial hypotheses.

Bibliography

[1] Bell, T. C., Cleary, J. G., and Witten, I. H. (1990). *Text compression*. Prentice-Hall, Inc.

[2] Bluche, T., Ney, H., and Kermorvant, C. (2013). Tandem hmm with convolutional neural network for handwritten word recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2390–2394.

[3] Breuel, T. M. (2017). Robust, simple page segmentation using hybrid convolutional mdlstm networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 733–740.

[4] Chen, K., Seuret, M., Hennebert, J., and Ingold, R. (2017). Convolutional neural networks for page segmentation of historical document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 965–970.

[5] Diem, M., Kleber, F., Fiel, S., Gruning, T., and Gatos, B. (2018). cbad: Icdar2017 competition on baseline detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1355–1360.

[6] Fawzi, A., Pastor, M., and Martínez-Hinarejos, C. D. (2017). Baseline detection on arabic handwritten documents. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, DocEng '17, pages 193–196, New York, NY, USA. ACM.

[7] Fernández-Mota, D., Lladós, J., and Fornés, A. (2014). A graph-based approach for segmenting touching lines in historical handwritten documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):293–312.

[8] Grüning, T., Leifert, G., Strauß, T., and Labahn, R. (2018). A two-stage method for text line detection in historical documents. *CoRR*, abs/1802.03345.

[9] Gruuening, T., Leifert, G., Strauss, T., and Labahn, R. (2017). A robust and binarization-free approach for text line detection in historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 236–241.

[10] Hermansky, H., Ellis, D. P. W., and Sharma, S. (2000). Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 3, pages 1635–1638 vol.3.

[11] Meier, B., Stadelmann, T., Stampfli, J., Arnold, M., and Cieliebak, M. (2017). Fully convolutional neural networks for newspaper article segmentation. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 414–419.

[12] Murdock, M., Reid, S., Hamilton, B., and Reese, J. (2015). Icdar 2015 competition on text line detection in historical documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1171–1175.

[13] Quirós, L. (2018). Multi-task handwritten document layout analysis. *CoRR*, abs/1806.08852.

[14] Romero, V., Toselli, A. H., and Vidal, E. (2012). *Multimodal Interactive Handwritten Text Transcription*. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing.

[15] Romero-Gómez, V., Toselli, A. H., Bosch, V., Sánchez, J. A., and Vidal, E. (2018). Automatic alignment of handwritten images and transcripts for training handwritten text recognition systems. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 328–333.

[16] Schone, P., Hargraves, C., Morrey, J., Day, R., and Jacox, M. (2018). Neural text line segmentation of multilingual print and handwriting with recognition-based evaluation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 265–272.

[17] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., and Alaei, A. (2013). Icdar 2013 handwriting segmentation contest. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1402–1406.

[18] Xu, Y., He, W., Yin, F., and Liu, C. (2017). Page segmentation for historical handwritten documents using fully convolutional networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 541–546.

CHAPTER 8

CONCLUSIONS

Chapter Outline

8.1 Introduction

In this chapter we present the global conclusions that can be drawn from the work presented in the thesis.

We will review our initial hypothesis and the objectives we decided on in order to prove our hypothesis. The way the different targets were met will be analysed and the specific scientific contributions created for each objective will be listed.

The work presented in this thesis has focused in advancing the document layout analysis field by means of adapting and applying machine learning techniques to it. To be more specific, we have mainly tackled the text region detection, classification and extraction problem by means of an stochastic framework. The text region issue is an important task that must be resolved in order to provide a truly open and digital access to manuscripts.

On one hand, this problem in its more coarse way of viewing it applies to the detection and classification of bigger page sections such as: paragraphs, diagrams, acts, music staffs, etc. The resolution of the text region issue in its more coarse facet is helpful in order to: provide better quality digital booklets, perform text searches as per specific regions and is a necessary step in top down document layout analysis approaches before text line segmentation is performed.

On the other hand, it can also be applied to a finer grain task like text line segmentation. Text line detection being a necessary prerequisite for some of the most mature solutions envisioned for handwritten text recognition problems. Text line segmentation is also important in order to prepare some of the higher quality digitalizations of texts and some of the richer user interfaces developed to allow experts to explore or review automatically generated transcriptions.

In this thesis we initially focused on how to solve the text region segmentation problem while trying to avoid heuristic based methods that included hard coded knowledge. In order to solve this we applied an stochastic framework that also lead us to providing a method to the generally non addressed issue of text region classification.

During this research we have had to review the state of the evaluation measures that were being used at the time. We needed to understand how they related to actual user effort when correcting the yielded results. The impact it had on the error of higher level tasks that depend on the detected and extracted regions also needed to be studied. We have had to develop our on error measures for both geometric error of baselines and for the classification of text lines.

We have created a dynamic programming algorithm that creates equidistant segmentation frontiers for text lines given a set of precomputed baselines.

We have adapted and applied a mature stochastic framework to the text region detection and classification problem. Furthermore, we have studied the use of language models as part of our framework reviewing how it allows experts to add valuable information and its overall impact in the errors defined.

We have performed several longitudinal studies reviewing how the technology developed applies to real production scenarios. Additionally, we have applied our methods to different corpus and a diverse type of regions scenarios: standard one column prose text, passing through notarial records, musical scores and royal charters. Finally, we have studied the benefits of using the information that can be extracted from word probabilistic indexes when tackling a problem were visual differences are not enough in order to perform text region classification.

This final chapter will conclude with a discussion on future research directions we intend to explore in order to extend the work presented in this thesis.

8.2 Scientific Outcomes

The different scientific outcomes presented in this thesis have been materialized as publications and contributions in research projects. Specifically 11 conference papers plus 1 journal paper (with an additional paper pending to be published) have been produced. Furthermore, the ideas, algorithms and systems developed in this thesis have been used in 6 research projects. In some cases the systems developed have actually been part of the final deliverables of certain projects. Next we will provide an overview of the different results obtained during this thesis and how they relate to the different scientific publications performed and research projects.

8.2.1 Evaluation Measures

Previously, evaluation measures for the text line segmentation problem consisted mainly in measuring how much of the expected contents (foreground pixels) of a text line were present in the extraction polygon or assigned to the required line [1–4]. Due to the difference in how our Stochastic Framework presented its results, as baselines and classification labels, we required new evaluation measures. This new evaluation measures were initially presented in our first conference paper:

- Bosch V., Toselli, A. H., and Vidal, E. (2012). Natural language inspired approach for handwritten text line detection in legacy documents. In *Proceedings of the 6th LaTeCH Workshop, pages 107–111. Association for Computational Linguistics*.

We expanded and used those evaluation measures throughout all the publications produced as part of this thesis. Additionally, we studied how those evaluation measures related to the effort required by a human in order to review and and correct the automatically yielded baselines. This review was performed as part of a longitudinal study performed over 40 pages as part of a production scenario in the Transcriptorium Project. The empirical test showed there was a high correlation between our *Relative Geometric Error* and the *User Review Time*. This outcome was published in:

- Bosch, V., Toselli, A. H., and Vidal, E. (2014). Semiautomatic text baseline detection in large historical handwritten documents. In *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 690–695*.

Although, the higher importance that the baseline detection phase had in comparison to the calculation of the extraction polygon on the actual text line segmentation was considered obvious, it had not been proven empirically. We performed experimentations in order to prove this. Furthermore, we flushed out some of the issues that the pixel based evaluation measure most used during a period of time had. Most glaring issue being the fact that no correlation between the *segmentation competition measure* and the *word error rate* used in handwritten text recognition could be seen. This work was published in:

- Romero V., Sánchez, J. A., Bosch, V., Depuydt, K., and de Does, J. (2015). Influence of text line segmentation in handwritten text recognition. In *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), pages 536–540*.

- Bosch V., Romero V., Toselli, A. H. and Vidal, E. (2018). Text line extraction based on distance map features and dynamic programming. In *16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 357–362*.

Additionally to all of the above work on evaluation measures, we have also provided in this thesis novel work on this subject that is yet to be published. In Sec. 7.3 of this thesis we reviewed if there was an actual relation between our *Relative Geometric Error* calculated over our yielded baselines and the *word error rate* of a handwritten text recognition task performed over the text line images extracted with those baselines. Results show that our evaluation measure based on the graphical difference between the yielded baselines and the groundtruth baselines correlate with the error obtained in the higher level task that are dependent on those baselines.

8.2.2 Text Line Detection and Classification

The main focus of the research presented in this thesis revolves around the idea of applying a stochastic framework to tackle the HTS problem. But Initially, this idea was only oriented to the subtask of text line segmentation. We developed our idea regarding the usage of Hidden Markov Models and Language Models and tested out the detection and classification performance of our approach in various corpus. Language models proved to be a very adequate way in order to automatically learn the underlying composition rules of the different document pages. The Language Models, which we renamed as Vertical Layout Models, allowed us to easily include knowledge provided by human experts to our machine learning approach.These results can be consulted in the following articles:

- Bosch V., Toselli, A. H., and Vidal, E. (2012). Natural language inspired approach for handwritten text line detection in legacy documents. In *Proceedings of the 6th LaTeCH Workshop, pages 107–111. Association for Computational Linguistics*.

- Bosch, V., Toselli, A. H., and Vidal, E. (2012). Statistical text line analysis in handwritten documents. In *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 201–206*.

Furthermore, in this thesis we have also documented work that is pending to be published in a Journal. Results regarding the usage of our method to record based documents and the application of a trainable bias-correction factor shown in Sec. 7.5 will be part of this publication.

8.2.3 Text Line Detection in Production Scenarios

We seldom see an actual study of how the developed technologies or techniques can actually be applied to real life production scenarios. Furthermore approaches based on machine learning techniques are usually seen as having a high entry cost for the actual benefits they provide. In order to prove the applicability of our approach we used the developed technology to yield groundtruth quality baselines as part of a transcription end-to-end system used for various corpus. The longitudinal studies in which we reviewed the applicability of our developed framework can be seen in seven articles.

The semi-automatic iterative process in order to apply our technology was first introduced in a seminal article were the framework was applied to the PROLOGO chapter of the first volume of the PLANTAS collection (presented in Sec. 5.7).

- Bosch, V., Toselli, A. H., and Vidal, E. (2014). Semiautomatic text baseline detection in large historical handwritten documents. In *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 690–695*.

In this initial study we were able to provide results that contradict the incorrect notion that machine learning based approaches require an initial large and costly amount of training data to yield adequate results in production scenarios. Furthermore we were also able to see how the system escalated correctly as it provided better quality text baselines when more data was made available to train the models that reside in its core. The study performed on the PROLOGO chapter was later extended to the rest of the first volume of the PLANTAS collection. This extended study was not only focused on the creation of ground-truth quality baselines, instead it covers the performance of the end-to-end handwritten text recognition system. This results were published in the following congress and journal articles:

- Bosch, V., Bordes-Cabrera, I., Muñoz, P. C., Hernández-Tornero, C., Leiva, L. A., Pastor, M., Romero, V., Toselli, A. H., and Vidal, E. (2014a). Computer-assisted transcription of a historical botanical specimen book: Organization and process overview. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, DATeCH Í4, pages 125–130, New York, NY, USA. ACM.*

- Toselli, A. H., Leiva, L. A., Bordes-Cabrera, I., Hernández-Tornero, C., Bosch, V., and Vidal, E. (2018). Transcribing a 17th-century botanical manuscript: Longitudinal evaluation of document layout detection and interactive transcription. In *Journal Digital Scholarship in the Humanities, 33(1):173–202.*

The implemented framework was partly developed for the *tranScriptorium* project. The use of the framework in the project with an example of actual practical usage of it with the HATTEM document (presented in Sec. 5.4) was published in the following article:

- Sánchez, J. A., Bosch, V., Romero, V., Depuydt, K., and de Does, J. (2014). Handwritten text recognition for historical documents in the transcriptorium project. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, DATeCH Í4, pages 111–117, New York, NY, USA. ACM.*

Our text line segmentation framework was also used as part of an end-to-end system to deliver production quality transcription in an interactive manner. The segmented lines yielded by our software were used as input for a computed assisted transcription system that allowed human experts to generate production quality transcriptions in a interactive and iterative manner. A study of this end-to-end system was performed using the RSEAPV corpus (presented in Sec. 5.8) to carry out the experiments. The results were published in the following article:

- Romero, V., Bosch, V., Hernánndez, C., Vidal, E., and Sánnchez, J. A. (2017). A historical document handwriting transcription end-to-end system. In *Pattern Recognition and Image Analysis Proceedings of the 8th Iberian Conference, IbPRIA 2017, pages 149–157, Springer International Publishing.*

Our framework was also used, in a fully automatic manner, as part of a system envisioned to generate training material for Handwritten Text Recognition Systems. HTR systems require a large amount of training data in order to train the optical models. The training data required for the optical model consists of pairs of text line images with their corresponding transcription. The developed system created this paired data by aligning the existing transcriptions of the document with the yielded text line systems of our framework. Results of this technique applied to the HATTEM document (presented in Sec. 5.4) can be read in the following article:

- Romero V., Toselli, A. H., Bosch, V., Sánchez, J. A., and Vidal, E. (2018). Automatic alignment of handwritten images and transcripts for training handwritten text recognition systems. In *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS), pages 328–333.*

Our last article regarding the use of machine learning approaches for text line segmentation was published in 2018. In this article the novel idea of applying Recurrent Neural Networks in a production scenario for line segmentation was applied. The longitudinal study was performed throughout the processing of 12 batches (containing 590 pages) of a recollection of notarial deeds. The study covered the transition for human annotated text lines to a layout analysis tool based on RNN.

- Quirós, L., Bosch, V., Serrano, L., Toselli, A. H., and Vidal, E. (2018). From HMMs to RNNs: Computer-assisted transcription of a handwritten notarial records collection. In *16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 116–121.*

8.2.4 Text Region Detection

As indicated before, the framework we developed was initially oriented to the subtask of text line segmentation. Regardless of this initial orientation, it can be applied in a straight forward manner to text region detection. Text region detection can be done both indirectly and directly. Indirectly by using the information yielded by our text line segmentation framework in order to compose the different page regions from it (as seen in Sec. 7.5) Or in a more direct manner by changing the optical models and compositional rules of our framework to directly tackle text regions. Our contribution to text region detection was performed using both approaches.

One of our region detection and classification contributions, that tackle the issue directly, was performed in the task of musical score layout detection. We adapted our method in order to differentiate between the different regions of a music score: blank, staff or lyrics that can be found in the different pages of the CAPITÁN corpus (presented in Sec. 5.9). The results of this application confirmed the potential of our technique to be directly applied for text region detection. This work is presented in:

- Bosch, V., Calvo-Zaragoza, J., Toselli, A. H., and Ruiz, E. V. (2016). Sheet music statistical layout analysis. In *Proceedings of 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 313–318.*

Moreover, to the work documented above regarding text region detection and classification, in this thesis we have presented results that are yet to be published. Determining the actual class of a

region might sometimes be impossible an impossible task to perform with just visual information. Any type of information regarding the textual contents of the region being analysed might improve the chances of classifying it correctly. In sec.7.8 of this thesis we provided information regarding the usage of word posterior-grams in order to enhance the performance of our region classifier. This technology was used successfully in order to be able to differentiate between the Acts documented in the CHANCERY corpus (presented in Sec. 5.10). The feature vectors enhanced with the information derived from these word probabilistic indexes allowed us to successfully determine whether the Acts were: starting, finishing or both starting and finishing in that same page.

8.2.5 Tandem: CNN and HMMs

The appearance of deep learning models have greatly impacted all the research fields. Although this change came at the end of the research that went into this thesis we still wanted to review the technology and the possible combinations it might have with out HMM based system.

In this thesis we have performed a tandem combination of a Convolutional Neural Network and our Hidden Markov models. This tandem makes use of the probability map yield by the CNN as input for out HMM framework. As seen in our experimentation, results also pending to be published, the ability the CNN has to flesh out the different text regions greatly outsurpasses any classical preprocess technique. In Sec. 7.9 we have also seen how the ability of the HMM to include prior information regarding the document which layout is being detected can help and improve the base CNN results. This should not come as a surprise as similar combinations have been performed both in ASR and HTR presenting the same positive impact.

8.3 Future Work

We finalize this chapter by identifying the future research topics and dimensions we intend to explore in order to extend the work presented in this thesis. Regarding text region segmentation and classification, there are avenues yet to explore. Although our current approach has been successfully applied to different document collections it is lacking in the way it tackles situations with more than one text region present in parallel: Side notes, indexes, two column text, etc. Our current approach is more one dimensional seeing a document page as a vertical concatenation of regions (Blank Space, Text, Diagrams, etc.) and thus is too simplistic to tackle complex layouts on its own. In order to tackle this issue studies regarding probabilistic graphical models must be performed in order to assess if there is an adequate way to represent and resolve this issue. Some prior experiments have proven that 2D Stochastic Grammars can be used to represent and tackle the document layout analysis problem [5]. Although the results are adequate, the method can be considered to be in its infant stage, further study is required to apply such a framework in a more general manner. It is important to note that our STRDC method could still be applied to further refine the solutions provided by these 2D systems. For example, the text line detection of the paragraphs detected by

these systems could be performed by our method. Also our system could be used to review and correct the segmentation and classification of the page regions present in the same column.

Another avenue to explore inside HTS is the level at which it is applied. As we saw in Chapter 2 current HTS techniques are only applied at two very distinct levels inside a single page of the given corpus. The idea of having two levels of information that must be tackled separately is something that should be re-evaluated. We believe that a method that can analyse all required levels of information for a single page in a concurrent manner should be sought after. Another self imposed restriction is the the physical level at which HTS approaches are currently applied. Most current methods can only be applied one page at a time which reduces the information available to provide an optimal result. We believe exploring ways in which to attack several pages, a chapter or even a whole book at the same time will provide performance gains.

Currently the community is putting much emphasis on the applicability of new Neural Networks (NNs) models for document layout analysis. These models were not covered extensively during the course of this thesis as the technology and software that have made them applicable came late during our research. Although current use of this technology has shown promising results it is important to note that these Neural Network Models do not actually perform text region segmentation and classification. In this thesis we covered the use of CNN and HMM in tandem mode. Although this simple combination has proven to be beneficial we believe that the more advanced combinations of these two models, seen in HTR, should be explored.

Finally, at the moment, there is not much study regarding the use of word probabilistic indexes in document layout analysis. Although the graphical layout might provide a large amount of information, on which to train classifiers or make decisions, it has its limits. As we have seen in the work presented in this thesis word probabilistic indexes can be used effectively to differentiate between text regions that are visually the same. The information that can be extracted from this automatically extracted probability maps can greatly enhance text region segmentation and classification. Further study on effective use of the word probabilistic indexes in combination with the different probabilistic models is required.

Bibliography

[1] Gatos, B., Antonacopoulos, A., and Stamatopoulos, N. (2007). Handwriting segmentation contest. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1284–1288.

[2] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2010). Icfhr 2010 handwriting segmentation contest. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 737–742.

[3] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2011). Icdar2009 handwriting segmentation contest. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1):25–33.

[4] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., and Alaei, A. (2013). Icdar 2013 handwriting segmentation contest. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1402–1406.

[5] Álvaro, F., Cruz, F., Sánchez, J.-A., Terrades, O. R., and Benedí, J.-M. (2015). Structure detection and segmentation of documents using 2d stochastic context-free grammars. *Neurocomputing*, 150:147 – 154. Bioinspired and knowledge based techniques and applications The Vitality of Pattern Recognition and Image Analysis Data Stream Classification and Big Data Analytics.