Statistical Text Line Analysis in Handwritten Documents*

Vicente Bosch
Inst. Tec. de Informática
Univ. Politécnica Valencia
Valencia - Spain
vbosch@iti.upv.es

Alejandro Héctor Toselli
Inst. Tec. de Informática
Univ. Politécnica Valencia
Valencia - Spain
ahector@iti.upv.es

Enrique Vidal
Inst. Tec. de Informática
Univ. Politécnica Valencia
Valencia - Spain
evidal@iti.upv.es

Abstract

In this paper we present an approach for text line analysis and detection in handwritten documents based on Hidden Markov Models, a technique widely used in other handwritten and speech recognition tasks. It is shown that text line analysis and detection can be solved using a more formal methodology in contraposition to most of the proposed heuristic approaches found in the literature. Our approach not only provides the best position coordinates for each of the vertical page regions but also labels them, in this manner surpassing the traditional heuristic methods. In our experiments we demonstrate the performance of the approach (both in line analysis and detection) and study the impact of increasingly constrained "vertical layout language models" on text line detection accuracy. Through this experimentation we also show the improvement in quality of the baselines yielded by our approach in comparison with a state-of-the-art heuristic method based on vertical projection profiles.

1. Introduction

Document Layout Analysis (DLA) is the process by which regions of interest in a document image are detected and categorized. This makes it an important and necessary task in any text recognition/transcription related tasks. Within DLA we find a Page Segmentation phase, an important initial step in charge of dividing the document image into homogeneous zones.

Text line analysis and detection (TLAD) is another DLA task embedded inside Page segmentation that constitutes an essential step in any modern text recognition and transcription systems that require input in the form of text line images. For example, text/image alignment [13], fully automatic handwritten text recognition [1, 14, 11] (HTR) or computer assisted transcription of text images (CATTI), where the users participate interactively in the actual transcription process [12]. Furthermore, due to the dependence such systems have on input quality, TLAD often has a significant impact on the final accuracy.

Text line detection in handwritten text entails a greater difficulty, in comparison with printed text lines, due to the inherit properties of handwritten text: variable inter-line spacing, overlapping and touching strokes of adjacent handwritten lines, etc.

Among the most popular state-of-the art methods for handwritten text line detection we find four main families: those based on (vertical) projection profiles [4], on the Hough transform [3], the repulse-attractive network approach [16] along with dynamic programming techniques to derive optimal paths between overlapping text lines [5].

It is worth noting that, most of the mentioned approaches somewhat involve heuristic adjustments of their parameters, which have to be properly tuned according to the characteristics of each task.

Our approach for TLAD in handwritten documents is based on Hidden Markov Models (HMM) trained with supervised data consisting of a set of detected and labelled text lines from the (kind of) documents considered. The seminal idea of our stochastic approach can be found in [6], where manually built HMMs and a fixed ergodic grammar are used in order to detect text lines in printed text. In this work we make (formal) use of HMMs to perform TLAD in handwritten text and perform a study on the impact of the language model in the detection accuracy.

We consider TLAD as a labelling process that assigns the same label to spatially aligned units such as pixels, connected components or characteristic points [4], and also actually finds and yields the physical locations of text lines in the image; More precisely, we look for the *baseline* position coordinates of hand-

*Work supported under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), MITTRAL (TIN2009-14633-C03-01) and also Univ. Politécnica Valencia (PAID-05-11)

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.274

written lines, which is defined as the fictitious straight line that follows and joins the lower part of the character bodies in a text line.

For the approach presented in this paper, we assume that page images or selected image regions contain only paragraphs of single-column (roughly) parallel text lines with no images or figures. Additionally, we assume that such images have been properly preprocessed so as to ensure that their text lines are roughly horizontal. These assumptions are adequate for the majority of handwritten documents of interest for transcription: large volumes with fairly good page structure.

In our experimentation, performed on legacy documents, we present results that show that the text line detection task can be addressed employing a more formal methodology as opposed to most of the proposed heuristic approaches found in the literature.

The rest of the paper is organized as follows. The next section we introduce the statistical framework and modelling scheme specially suited for TLAD. Then the system architecture is explained in detail in section 3. Section 4 presents the experimental set-up and results and, finally, section 5 summarizes the work presented and draws preliminary conclusions and directions for future research.

2. TLD Statistical Framework

Similarly to how the statistic framework of automatic speech and handwritten text recognition (ASR,HTR) is established, the handwritten text line detection problem can be also formulated as the problem of finding a most likely text line sequence hypothesis, $\hat{\mathbf{h}} = \langle h_1, h_2, \ldots, h_n \rangle$, for a given handwritten page image (or selected region image) represented by an observation sequence[1] $\mathbf{o} = \langle o_1, o_2, \ldots, o_m \rangle$, that is:

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{o}) \tag{1}$$

Using the Bayes' rule we can decompose the probability $P(\mathbf{h} \mid \mathbf{o})$ into two terms:

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} P(\mathbf{o} \mid \mathbf{h}) \cdot P(\mathbf{h}) \tag{2}$$

In the jargon of ASR or HTR these terms represent the morphological and syntactic knowledge level respectively, where $P(\mathbf{o} \mid \mathbf{h})$ is typically approximated by HMMs, while $P(\mathbf{h})$ by an N-gram language model (LM).

[1] Henceforward, in the context of this formal framework, each time it is mentioned image of *page or selected text*, we are implicitly referring to its input feature vector sequence "$\mathbf{o}$" describing it. See details in section 3.2.

In this work, we are interested not only in detecting and labelling the text lines in a given (page) image, but also their exact physical locations on it. In this sense, by solving Eq. (2), such physical locations are determined by the optimal subsequences of $\mathbf{o}$ aligned with each of the detected text lines $h_1, h_2, \ldots, h_n$. These optimal subsequences are implicit or "hidden" in Eq. (2), which can be rewritten as:

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} \sum_{\mathbf{b}} P(\mathbf{o}, \mathbf{b} \mid \mathbf{h}) \cdot P(\mathbf{h})$$

where $\mathbf{b}$ is an *alignment*; that is, an ordered sequence of $n+1$ marks $\langle b_0, b_1, \ldots, b_n \rangle$, used to demarcate the subsequences belonging to each text line. The marks b_0 and b_n always point out to the first and last components of $\mathbf{o}$ (see Fig. 1). Now, approximating the sum in (3) by

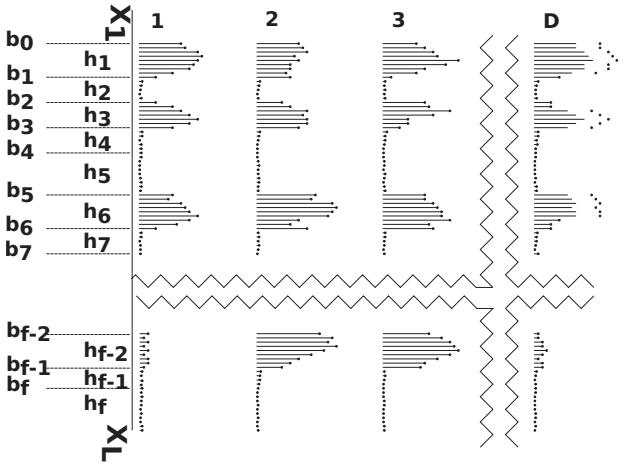

Figure 1. Schematic representation for a page image extraction of L feature vectors of D components. See details in section 3.2.

the dominant term, $\max_{\mathbf{b}} P(\mathbf{o}, \mathbf{b} \mid \mathbf{h})$:

$$(\hat{\mathbf{b}}, \hat{\mathbf{h}}) \approx \arg\max_{\mathbf{b},\mathbf{h}} P(\mathbf{h}) \cdot P(\mathbf{o}, \mathbf{b} \mid \mathbf{h}) \tag{3}$$

where $\hat{\mathbf{b}}$ is the optimal alignment. Eq. (3) can be expanded to,

$$(\hat{\mathbf{b}}, \hat{\mathbf{h}}) = \arg\max_{\mathbf{b},\mathbf{h}} P(\mathbf{h}) \cdot P(o_{b_0}^{b_1} \mid \mathbf{h})\, P(o_{b_1}^{b_2} \mid o_{b_0}^{b_1}, \mathbf{h}) \ldots P(o_{b_{n-1}}^{b_n} \mid o_{b_0}^{b_{n-1}}, \mathbf{h}) \tag{4}$$

Assuming that each subsequence $o_{b_{i-1}}^{b_i}$ is independent from $o_{b_0}^{b_1}, \ldots, o_{b_{i-2}}^{b_{i-1}}$, and it also depends only of h_i, Eq. (4) can be rewritten as,

$$(\hat{\mathbf{b}}, \hat{\mathbf{h}}) \approx \arg\max_{\mathbf{b},\mathbf{h}} P(\mathbf{h}) \cdot P(o_{b_0}^{b_1} | h_1) \ldots P(o_{b_{n-1}}^{b_n} | h_n) \tag{5}$$

Which is optimally solved by using the Viterbi search algorithm [2].

2.1. Modelling

Our line detection approach is based on two modelling levels: morphological and syntactical. The morphological level, expressed as the $P(\mathbf{o} \mid \mathbf{h})$ term (see Eq.(2) and Eq.(5)), is modelled by using HMMs and is in charge of explaining the different vertical line regions classes that appear on the input images along its vertical direction. In our line detection approach four different kinds of vertical regions are defined:

Normal text Line-region (NL): Region occupied by the main body of a normal handwritten text line.

Inter Line-region (IL): Defined as the region found within two consecutive normal text lines, characterized by being crossed by the ascenders and descenders belonging to the adjacent text lines.

Blank Line-region (BL): Large rectangular region of blank space usually found at the start and end of page image (top and bottom margins).

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

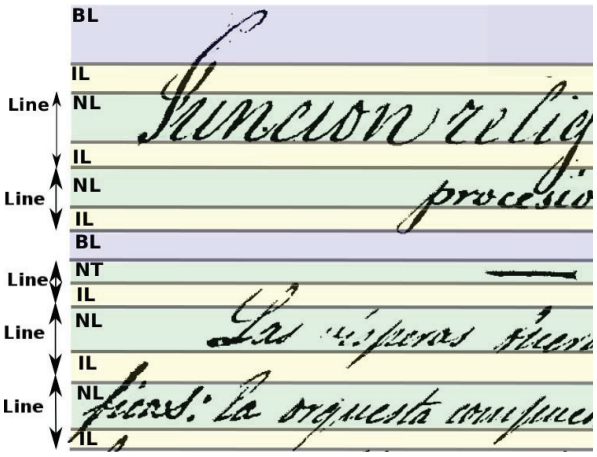

Figure 2. Examples of the different classes of line regions.

We model each of these regions by an HMM which is trained with instances of such regions. Basically, each line-region HMM is a stochastic finite-state device that models the succession of feature vectors extracted from instances of the specific line-region images. In turn, each HMM state generates feature vectors following an adequate parametric probabilistic law; typically a mixture of Gaussian densities. The adequate number of states and Gaussians per state may be conditioned by the available amount of training data.

Once an HMM "topology" (number of states and structure) has been adopted, the model parameters can be easily trained from instances (sequences of features vectors) of full images containing a sequence of line-regions (without any kind of segmentation) accompanied by the reference labels of these images that correspond to the actual sequence of line-region classes. This training process is carried out using a well known instance of the EM algorithm called forward-backward or Baum-Welch re-estimation [2].

The syntactic modelling level , expressed as the $P(\mathbf{h})$ term (see Eq.(2) and Eq.(5)), is responsible for defining the way that the different line regions can be concatenated in order to produce a valid page structure. It is worth noting that at this level, NL and NT line regions are always forced to be followed by IL region: NL+IL and NT+IL. We can also use the LM to impose restrictions about the minimum or maximum number of line-regions to be detected. The LM for our text line detection approach, is implemented as a stochastic finite state grammar (SFSG) which recognizes valid sequences of elements (line regions): NL+IL, NT+IL and BL.

Both modelling levels, the morphological and syntactical, which are represented by finite-state automaton, can be integrated into a single global model on which Eq. (5) is easily solved; that is, given an input sequence of raw feature vectors, an output string of recognized sequence of line region class labels along with their corresponding coordinate positions are obtained. It should be mentioned that, in practice, HMM and LMs probabilities (usually expressed in logarithms) are generally "balanced" before being used in Eq. (5). This is carried out by using a *"Grammar Scale Factor"* (GSF), the value of which is tuned empirically.

3. System Architecture

The flow diagram of Fig. 3 displays the overall process of the proposed handwritten text line analysis and detection approach.

It is composed of four different phases: image preprocessing, feature extraction, HMMs and LM training and decoding. Next we will overview the first two phases, preprocessing and feature extraction, since the rest has already been covered in the preceding section.

3.1. Preprocessing Module

Background removal and noise reduction is performed on the input images by applying a bi-dimensional median filter. The resulting images skew

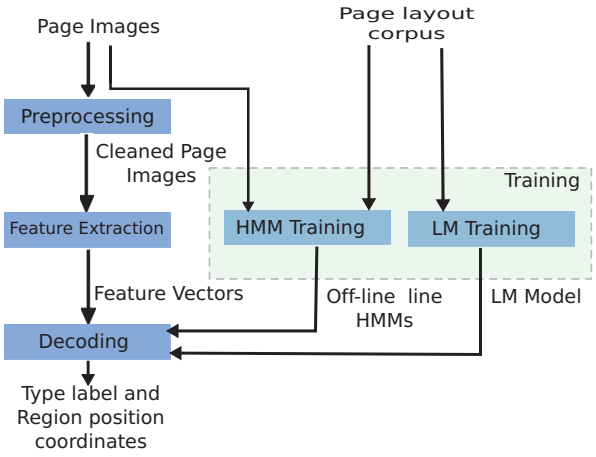

Figure 3. Global scheme of the handwritten text line detection process.

is also corrected by applying projection profile and RLSA [15], along with standard techniques to calculate the skew angle.

3.2. Feature Extraction Module

Since our TLAD approach is based on HMMs, each preprocessed image must be represented as a sequence of feature vectors. This is done by dividing the already preprocessed image into D non-overlapping rectangular regions (from left-to-right) with height equal to the image-height L (see Fig. 4).

In each of these rectangular regions, we compute the grey level histogram of the horizontal projection, after applying RLSA to obtain a more emphasized vertical profile of the horizontal projection, which we will refer to from now on as the "vertical projection profile". Finally, to eliminate local maxima on the obtained vertical projection profiles, they are smoothed with a rolling median filter [7]. The resulting effect of this process is shown in Fig. 4.

In this way, a D-dimensional feature vector is constructed for each page/block image row of pixels, by stacking the D projection profile values corresponding to that row. Hence, at the end of this process, a sequence of L D-dimensional feature vectors is obtained (see Fig. 1).

4. Experimental Setup and Results

In order to study the effectiveness of our proposed line detection approach, different experiments were carried out. We are mainly interested in assessing the impact upon final text line detection accuracy of employing increasingly restrictive LMs and compare our results with other text line detection methods currently in use.

4.1. Corpus Description

Experiments were carried out using a corpus compiled from a XIX century Spanish manuscript identified as "Cristo-Salvador" (CS), which was kindly provided by the *Biblioteca Valenciana Digital* (BiVaLDi)[2]. This is a rather small document composed of 53 color images of text pages, scanned at 300 dpi and written by a single writer. Some page images examples are shown in Fig. 5.

Figure 5. Examples of page images from CS corpus.

In this case. we employ the already predefined *book* partition [9]. This partition divides the data-set into a test set containing 20 page images, and a training set composed of the 33 remaining pages. Table 1 presents basic statistical information of the *book* partition.

Table 1. Basic statistics of the Cristo-Salvador corpus "book" partition.

Number of:	Training	Test	Total
Pages	33	20	53
Normal-text lines (NL)	685	497	1 182
Blank Lines (BL)	73	70	143
Non-text Lines (NT)	16	8	24
Inter Lines (IL)	701	505	1 206

Each page was annotated with a succession of reference labels (NL, NT, BL and IL) indicating the kind of line-regions it is composed of. Line positions were obtained in a first instance by executing standard methods for text line detection based on the whole-line vertical projection profile, which were afterwards manually labelled, verified, adjusted and/or rectified by a human operator to ensure correctness.

[2] http://bv2.gva.es.

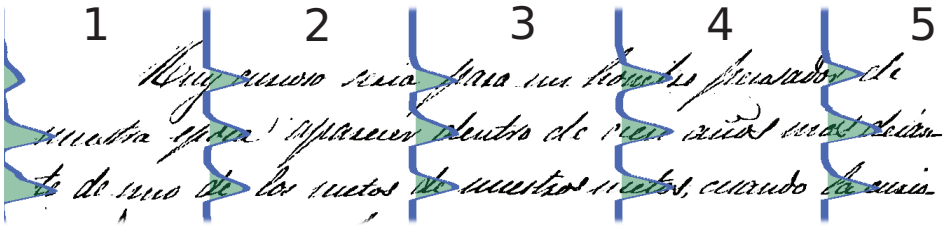

Figure 4. Partial page image visualization of 5 ($D = 5$) rectangular regions across over 3 handwritten text lines. For each region, its vertical projection profile is also plotted.

4.2. Models and Baseline

For our experimentation we considered the following language models: the *prior* (PRI) and *conditional* (CND) represented by topological different SFSGs. The PRI LM transition probabilities are estimated from the training set as the fraction of the number of appearances of each line region label over the whole count of labels. The CND LM also considers the previous line region label in order to perform the estimation. These estimates somewhat resemble the uni-gram and bi-gram LMs calculations, except no smoothing strategy is implemented here. Additionally, we defined for each test page a *line-number constrained* LM (LN-C), which also uses the CND LM probabilities to populate the model, that enforces a total number of possible line-regions (line or blank space) to detect as per the number of reference line-region labels of that test page. LN-C is conceived for its utilization in (parts of) documents or document collections that present a homogeneous number of lines per page.

For comparison purposes, we obtained a baseline result by employing a standard, already implemented line detection approach based on plain whole-line vertical projection profiles [10]. The method requires as input the expected number of text lines to be found on the corresponding page and yields the found baseline coordinates but does not output any labelling information.

The experimentation was performed on the described CS corpus using a simple hold-out validation as pet the CS "book" partition. Initially some parameters were set up: feature extraction dimension D, HMM topology (number of states and Gaussians), number of Baum-Welch iterations, decoding grammar scale factor (GSF) and word insertion penalty (WIP). Through some informal experimentation adequate values were found: feature vector dimension set up to 2, employing left-to-right HMM topologies with 4 states and 32 Gaussian mixtures per state trained by running 3 cycles of Baum-Welch re-estimation algorithm. The remaining parameters related with the decoding process itself (GSF and WIP), were tuned also to obtain the best results for each of the models.

4.3. Evaluation Measures

In order to assess the quality of the proposed TLAD, two kinds of measures have been adopted: "line error rate" (LER), considered a qualitative measure, is calculated as the number of incorrectly assigned line labels divided by the total correct line regions; and the "alignment accuracy rate" (AAR) which, addressing the evaluation more from a quantitative point of view, measures the geometrical accuracy of the detected line region positions with respect to the corresponding (correct) reference marks.

The LER is obtained by comparing the sequences of automatically obtained region labels($\hat{h}$ in eq. 5) with the corresponding sequences. This is computed in the same way as the the well known WER, with equal editing-costs assigned to deletions, insertions and substitutions [8].

Concerning AAR computation, it is carried out in two phases. First, for each page, we find the best alignment between the system-proposed line position marks ($\hat{b}$ in eq. 5) and the page reference marks by minimizing its accumulate absolute differences using dynamic programming. Then, in the second phase, taking into account only the detected regions marked as NL, the AAR is calculated as the average value (and standard deviation) of the already computed absolute differences of the position marks for all the pages globally.

4.4. Results

In Table 2 we report the best figures for LER and ARR achieved through the indicated experimentation process for the three LMs and the heuristic baseline method. The AAR mean and std-dev are given in this case in percentage of the text line average width (80 pixels).

Although the HEUR method does not formally use a LM it requires as input the number of text lines (NL) present in the page thus for comparisson reasons we consider it to be using a LN-C model.

Table 2. Best figures of LER(%) and AAR(%) obtained for our statistical text line analysis approach (STLAD) and the heuristic one (HEUR), using different kind of language models: Prior (PRI), Conditional (CND) and Line-Number Constrained (LN-C).

Approach	LM	LER(%)	AAR(%)	
			mean	std-dev
STLAD	PRI	0.86	9.04	15.91
	CND	0.70	8.81	15.40
	LN-C	0.34	8.75	13.15
HEUR	LN-C	–	9.94	29.84

As can be seen in Table 2, the more restrictive the LM is, the better accuracy is achieved. Similarly the quantitative evaluation shows that more construed LMs provide better baseline coordinate hypotheses (closer to the ground truth ones). In the case of HEUR LM, the obtained AAR (std) is not as good as the STLAD's with a much higher std-dev.

5. Conclusions

We have shown a new way of addressing text line analysis and detection by using a statistical framework, similar to the already employed in many popular NLP tasks, that avoids the traditional heuristics approaches generally used for this issue.

In the future our intention is to extend this approach to perform a more diverse classification of line-regions: titles, short lines, beginning and and end of a paragraphs, etc; which would allow us to perform logical layout analysis. Furthermore, it is envisioned that the proposed stochastic framework serves as a cornerstone to allow user interactivity in the line detection process.

References

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

[2] F. Jelinek. *Statistical methods for speech recognition*. MIT Press, 1998.

[3] L. Likforman-Sulem, A. Hanimyan, and C. Faure. A hough based algorithm for extracting text lines in handwritten documents. *Document Analysis and Recognition, International Conference on*, 2:774, 1995.

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

[5] M. Liwicki, E. Indermuhle, and H. Bunke. On-line handwritten text line detection using dynamic programming. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 447 –451, sept. 2007.

[6] Z. Lu, R. Schwartz, and C. Raphael. 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, 2000.

[7] R. Manmatha and N. Srimal. 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, 1999. Springer-Verlag.

[8] I. A. McCowan, D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner, and H. Bourlard. On the use of information retrieval measures for speech recognition evaluation. Idiap-RR Idiap-RR-73-2004, IDIAP, Martigny, Switzerland, 0 2004.

[9] V. Romero, A. H. Toselli, L. Rodríguez, and E. Vidal. 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), August 2007.

[10] I. Sánchez-Cortina, N. Serrano, A. Sanchis, and A. Juan. A prototype for interactive speech transcription balancing error and supervision effort. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 325–326, 2012.

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

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

[13] A. H. Toselli, V. Romero, and E. Vidal. *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,, 2011. Caroline Sporleder, Antal van den Bosch y Kalliopi Zervanou (Eds.).

[14] A. Vinciarelli, S. Bengio, and H. Bunke. 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, june 2004.

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

[16] E. Öztop, A. Mülayim, V. Atalay, and F. Yarman-Vural. Repulsive attractive network for baseline extraction on document images. *Signal Processing*, 75(1):1 – 10, 1999.