UNIVERSIDAD POLITÉCNICA DE VALENCIA

MASTER IN ARTIFICIAL INTELLIGENCE
PATTERN RECOGNITION AND DIGITAL IMAGE

Multiple Contributions to Interactive Transcription and Translation of Old Text Documents

Work
presented by Nicolás Serrano Martínez Santos
supervised by Dr. Alfons Juan Císcar

November 30, 2009

Acknowledgements

Work supported by the EC (FEDER/FSE) and the Spanish MCE/MICINN under the MIPRCV "Consolider Ingenio 2010" programme (CSD2007-00018), the iTransDoc project (TIN2006-15694-CO2-01) and the FPU scholarship AP2007-02867.

CONTENTS

CHAPTER 1

Introduction

There are huge historical document collections residing in libraries, museums and archives that are currently being digitized for preservation purposes and to make them available worldwide through large, on-line digital libraries. The main objective, however, is not to simply provide access to raw images of digitized documents, but to annotate them with their real informative content and, in particular, with text transcriptions and, if convenient, text translations too. This work aims at contributing to the development of advanced techniques and interfaces for the analysis, transcription and translation of images of old archive documents, following an interactive-predictive approach. Our hypothesis is that this goal cannot be reliably accomplished by fully automatic techniques; instead, a person-machine collaborative model has to be followed so as to produce accurate document interpretation in a cost-effective way. In order to show this hypothesis, a software tool has been developed and evaluated. It must be noted that the work reported here has been carried out within the framework of the Spanish research project "Interactive Transcription and Translation of Old Text Documents (iTransDoc)" [1].

More specifically, the contributions described in this work are the following:

GERMANA & RODRIGO: Preparation of databases.

Annotation of digitized pages from historical document collections is very important to research on automatic extraction of text blocks, lines, and handwriting recognition. However, there is a lack of databases of annotated old text documents. In this work, we have collaborated in the preparation of two databases of old text documents: GERMANA & RODRIGO. On the one hand, GERMANA is the result of digitizing and annotating a 764-page Spanish manuscript from

1891, in which most pages only contain nearly calligraphed text written on ruled sheets of well-separated lines. GERMANA is solely written in Spanish up to page 180. However, it many parts that are written in languages different from Spanish, namely Catalan, French and Latin. On the other hand, RODRIGO is the result of digitizing and annotating a manuscript from 1545 entitled *"Historia de España del arçobispo Don Rodrigo",* and completely written in old Castilian (Spanish) by a single author. It is a 853-page bound volume, in which most pages only a single text block of nearly calligraphed handwriting on well-separated lines. Both, GERMANA and RODRIGO have been made publicly available on-line, and are described in two articles in international conferences:

- **ICDAR-2009:** D. Pérez, L. Tarazón, **N. Serrano**, F. Castro, O. Ramos and A. Juan. The GERMANA database. *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR 2009).* Barcelona (Spain). July 2009.
- **LREC-2010: N. Serrano**, F. Castro and A. Juan. The RODRIGO database. *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010).* Valletta (Malta). May 2010. (Submitted)

GIDOC: Gimp-based Interactive transcription of old text DOCuments.
In accordance with the interactive-predictive approach described above, a system prototype called GIDOC has been developed to provide user-friendly, integrated support for layout analysis, line detection and handwriting transcription. This work has led to four publications in international conferences:

- **ICIAP-2009:** L. Tarazón, D. Pérez, **N. Serrano**, V. Alabau, O. Ramos Terrades, A. Sanchis and A. Juan. Confidence Measures for Error Correction in Interactive Transcription of Handwritten Text. *Proceedings of the 15th International Conference on Image Analysis and Processing (ICIAP 2009).* Vietri sul Mare (Italy). September 2009.
- **DRR-2010:** O. Ramos, **N. Serrano** and A. Juan. Interactive-predictive detection of handwritten text blocks. *Proceedings of the 17th Document Recognition and Retrieval Conference (DRR 2010).* San Jose (USA). January 2010. (Accepted)
- **WEBIST-2010: N. Serrano**, L. Tarazón, D. Pérez, O. Ramos-Terrades and A. Juan. The GiDOC Prototype. *Proceeding of 6th International Conference on Web Information Systems and Technologies (WEBIST 2010).* Valencia (Spain). April 2010. (Submitted)
- **CHI-2010: N. Serrano** and A. Juan. Demonstration of the GiDOC Prototype. *Media Showcase of 28th ACM Conference on Human Factors in Computing Systems (CHI 2010).* Atlanta (USA). April 2010. (Submitted)

Adaptation and interaction in handwriting recognition.
Using a framework based on the interactive-predictive approach, the successively produced transcriptions can be used to better adapt the system to the task.

However, if transcriptions are only partially supervised, then recognition errors may go unnoticed to the user and have a negative effect on model adaptation. We study the effect of establishing a fixed degree of supervision and we propose a simple yet effective method to find an optimal balance between recognition error and supervision effort. This work has led to two publications in international conferences:

- **ICMI-MLMI-2009: N. Serrano**, D. Perez, A. Sanchís and A. Juan. Adaptation from Partially Supervised Handwritten Text Transcriptions. *In Proceedings of the 11th International Conference on Multimodal Interfaces and the 6th Workshop on Machine Learning for Multimodal Interaction (ICML-MLMI 2009).* Cambridge MA, USA. September 2009.
- **IUI-2010: N. Serrano**, A. Sanchis and A. Juan. Balancing Error and Supervision Effort in Interactive-Predictive Handwriting Recognition. *Proceedings of 14th Intelligent User Interface (IUI 2010).* Hong-Kong, China. February 2010. (Accepted)

Kernel regression approach to machine translation.

Due to its difficulty, several authors have approached automatic translation as a statistical pattern recognition problem. However, we present a novel machine translation framework based on Kernel Regression techniques. The translation process is modeled as a string-to-string mapping. This translation mapping is learnt by linear regression. Once the target feature vector is obtained, we use a multi-graph search to find the translated sentence. This work has led to a publication in an international conference:

- **IbPRIA-2009: N. Serrano**, J. Andrés-Ferrer and F. Casacuberta. On a Kernel Regression Approach to Machine Translation. *Proceedings of the 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2009).* Póvoa de Varzim, Portugal. June 2009.

It must be noted that the contributions described above are the result of a collaborative work involving other authors and, in particular, authors that are also presenting their Master's Theses for the "Master in Artificial Intelligence, Pattern Recognition and Digital Image". Nevertheless, when comparing the work reported here to that in these other Master's Theses, the author of this Thesis should be considered as the leading author of the work reported in the articles denoted above as LREC-2010, WEBIST-2010, DRR-2010, CHI-2010, ICMI-MLMI-2009, IUI-2010 and IbPRIA-2009. Accordingly, the work reported in these articles is described with full detail in Chapters 2 (LREC-2010), 3 (DRR-2010, WEBIST-2010 and CHI-2010), 4 (ICMI-MLMI-2009), 5 (IUI-2010) and 6 (IbPRIA-2009). On the other hand, the work reported in the remaining articles is briefly described in Chapters 2 (ICDAR-2009) and 4 (ICIAP-2009). The reader is referred to the Master's Thesis by D. Pérez [2] and L. Tarazón [3] for more details on the ICDAR-2009 and ICIAP-2009 articles, respectively. Also, it must be noted that the basic GIDOC prototype has been developed by the author of this Thesis together with D. Pérez and L. Tarazón, at the same level of dedication effort, and thus all these three authors should be considered as leading authors of the WEBIST-2010 article.

For the sake of clarity, the correspondence between Chapters and articles is summarized in Table 1.1, together with conference quality indicators (CORE rank).

Article	Status	Quality indicators	Contribution	Chapter
ICML-MLMI 2009	Published	CORE B	Leading author	4
IBPRIA-2009	Published	Lecture Notes in CS	Leading author	6
IUI-2010	Accepted	CORE A	Leading author	5
DRR-2010	Accepted	-	Leading author	3.2
ICDAR-2009	Published	CORE A	Co-author	2.1
ICIAP-2009	Published	CORE A	Co-author	4.1
CHI-2010	Submitted	CORE A+	Leading author	3
WEBIST-2010	Submitted	CORE C	Leading author	3
LREC-2010	Submitted	CORE C	Leading author	2.2

Table 1.1: Articles generated from the work described in this document.

CHAPTER 2

Databases for Handwritten Text Recognition

Annotation of digitized pages from historical document collections is very important to research on automatic extraction of text blocks, lines, and handwriting recognition. However, there is a lack of databases of annotated old text documents. We present these new handwritten text database, GERMANA and RODRIGO, to facilitate empirical comparison of different approaches to automatic extraction of text blocks, lines, and handwriting recognition.

2.1 The GERMANA Database

In this section, we present a handwritten text database, GERMANA. GERMANA is the result of digitising and annotating a 764-page Spanish manuscript entitled *"Noticias y documentos relativos a Doña Germana de Foix, última Reina de Aragón"* and written in 1891 by Vicent Salvador, the Cruïlles' marquis. It has approximately $21K$ text lines manually marked and transcribed by palaeography experts. For a detailed description of this database refer to [2].

2.1.1 Description

GERMANA is not a particularly difficult task for several reasons. First, it is a single-author book on a limited-domain topic: the life of *Germana de Foix* (1488-1538), niece of King Louis XII of France and second wife of Ferdinand the Catholic of Aragon. Also, the original manuscript was well-preserved and most pages only contain nearly calligraphed text written on ruled sheets of well-separated lines. Moreover, the manuscript comprises about $217K$ running words from a vocabulary of $30K$ words which, apparently, is a reasonable amount of data for single-author handwriting and language modelling.

It goes without saying that text line extraction and off-line handwriting recognition on GERMANA is not, by contrast, particularly easy. GERMANA has typical characteristics of historical documents that make things difficult: spots, writing from the verso appearing on the recto, unusual characters and words, etc. Also, the manuscript includes many notes and appended documents that are written in languages different from Spanish, namely Catalan, French and Latin.

All in all, we think that GERMANA entails an appropriate trade-off between task complexity and amount of data. To our knowledge, it is the first publicly available database for handwriting research, mostly written in Spanish and comparable in size to standard databases such as IAM [4, 5]. Due to its sequential book structure, it is also well-suited for realistic assessment of *interactive* handwriting recognition systems [6]. Moreover, it can be used as well to test approaches for language identification and adaption from single-author handwriting.

2.1.2 The database

The manuscript was carefully scanned by experts from the Valencian Library at 300dpi in true colours. As with historical documents in general, scanned pages have noise effects like spots, tears, ink fading and transparency of back side. Also, they show a slight warping due to book binding. Nevertheless, the manuscript can be easily read and thus we decided not to apply any preprocessing to it for the purpose of annotating ground-truth.

Ground-truth annotation of GERMANA consisted of two parts. On the one hand, all text blocks were marked with minimal enclosing rectangles and, within each text block, each text line was marked by its (straight) baseline. This was done semi-automatically by means of the GiDOC prototype (for a detailed description refer to Chapter 3) we developed specifically for block and line annotation of GERMANA. All blocks and baselines detected automatically were also manually supervised, and corrected when needed.

Table 2.1 contains some basic statistics drawn from our GERMANA transcription. Note that the Spanish part of GERMANA comprises about $17K$ text lines and $177K$ running words from a lexicon of $20K$ words. It is also worth noting that 56% of the words only occur once (singletons). Regarding the other, non-Spanish parts, it is clear that they are not large enough to reliably estimate independent models for them . Instead, it would be very interesting to see how models trained with different data can be adapted to them. In particular, character HMMs trained with the Spanish part might be very well reused without significant changes.

The database is available at the PRHLT website (`prhlt.iti.es`) for non-commercial research.

Lang.	Pages	Lines	Words (K)	Lexicon Size (K)	Lexicon Sing. (%)	Char set
Spanish	595	16599	176.8	19.9	55.6	111
Catalan	87	2417	26.9	4.6	63.2	86
Latin	29	951	8.3	3.4	69.2	87
French	8	266	3.0	1.1	71.1	82
German	8	228	1.5	0.6	52.7	71
Italian	2	68	0.8	0.3	67.3	59
None	35	0	0.0	0.0	0.0	0
All	764	20529	217.2	27.1	57.4	115

Table 2.1: Basic statistics of GERMANA (Sing=Singletons, words occurring only once).

2.2 The RODRIGO database

In this section, we present another handwritten text database, which will be referred to as RODRIGO. In this case, we have selected a manuscript much older than that of GERMANA, from 1545, which is publicly available in digitized form, at 300dpi in true colors, from the Spanish "Ministerio de Cultura" web site [7]. The original manuscript is a 853-page bound volume, entitled *"Historia de España del arçobispo Don Rodrigo"*, and completely written in old Castilian (Spanish) by a single author. We carefully annotated all text blocks, lines and transcriptions, resulting in approximately $20K$ lines and $231K$ running words from a lexicon of $17K$ words, that is, very similar to GERMANA in size. The main purpose of this work is to let this annotation known to researchers and to provide an adequate reference for future studies. The interested reader can download it from [8].

As GERMANA, RODRIGO is not a particularly difficult task for text and block line detection since most pages only contain a single text block of nearly calligraphed handwriting on well-separated lines. It is also a single-author manuscript on a limited-domain task and, easier than GERMANA, it is only written in Spanish. Nevertheless, RODRIGO comes from a much older manuscript, and thus the typical difficult characteristics of historical documents are more evident. In particular, the writing style, which has clear Gothic influences, is significantly more complex than that of GERMANA.

2.2.1 The manuscript

As said above, the RODRIGO database corresponds to a manuscript from 1545 entitled *"Historia de España del arçobispo Don Rodrigo"*, and completely written in old Castilian (Spanish) by a single author. It is a 853-page bound volume divided into 307 chapters describing chronicles from the Spanish history. Most pages only a single text block of nearly calligraphed handwriting on well-separated lines. This can be seen in Fig. 2.1, where it is also apparent that writing style has clear Gothic influences [9].

Other characteristic details of RODRIGO that can be clearly appreciated in Fig. 2.1 are:

- The author tends to embellish the writing, specially in broad white spaces, resulting in the extension of some ascenders and descenders across whole words.

- Natural blank spaces between successive words are often omitted; e.g., the words “de la” are written as a single word “dela” in the third line from the bottom of the page shown in Fig. 2.1. Sometimes, on the contrary, artificial blank spaces are inserted within a single word; e.g., the word “llegaronse” is written as two words, “llegaron se”.

- Each chapter should begin with a dropcap, but the manuscript contains no dropcaps, probably because it was never brought to an artist to do so. Instead, there is a blank area in each position where a dropcap should have been inserted and, in most cases, the corresponding letter is written in small size (see Fig. 2.1).

2.2.2 The database

The manuscript was carefully digitized by experts from the Spanish *Ministry of Culture,* at 300dpi in true colors, and it is publicly available at [7]. As with historical documents in general, scanned pages have noise effects like spots, tears, ink fading and transparency of back side. Also, they show a slight warping due to book binding. Nevertheless, the manuscript can be easily read and thus we decided not to apply any preprocessing (apart from de-saturation) to it for ground-truth annotation.

We followed an annotation procedure very similar to that used for the GERMANA database [10]. First, all text blocks were annotated with minimal enclosing rectangles and, within each text block, each text line was marked by its (straight) baseline. This was done semi-automatically by means of the GiDOC prototype (for a detailed description refer to Chapter 3). All blocks and baselines automatically detected were also manually supervised, and corrected when needed.

On the other hand, the whole manuscript was transcribed line by line, by a palaeography expert, in accordance with the following transcription rules:

- Page and line breaks are copied exactly.
- Missing natural blank spaces between successive words are indicated by the symbol “‿”.
- Inserted artificial blank spaces within words are indicated by the symbol “␣”.
- No spelling mistakes are corrected.
- No case or accentuation change is done.
- Punctuation signs are copied as they appear.

- Word abbreviations are first copied verbatim, except for sub-indices and super-indices, which are written in LaTeX-like notation as `_{sub}` and `^{super}`, respectively. Then, they are followed by the corresponding word between brackets. Thus, for instance, `q`$^{\texttt{i}}$`er.` is transcribed as `q^{i}er[quier]`.

- The symbol "$" is appended to each line having a broken word at its end.

The total time required for a single expert to manually transcribe the whole manuscript was estimated as 500 hours; that is, approximately 35 minutes per page on average.

The complete annotation of RODRIGO is publicly available, for non-commercial use, at [8]. It comprises about $20K$ text lines and $231K$ running words from a lexicon of $17K$ words, which is comparable in size to standard databases such as IAM [4, 5]. Approximately, 53% of the words only occur once (singletons). To compute these statistics,punctuation signs were isolated and abbreviations were substituted by their corresponding words.

2.2.3 Experiments

As discussed in this chapter introduction, RODRIGO is introduced to facilitate comparison of different approaches to automatic extraction of text blocks, lines, and handwriting recognition. In this section, however, we will restrict ourselves to (automatic) transcription (handwriting recognition). More specifically, our aim is simply to provide baseline results for reference in future studies, using standard techniques and tools; i.e., HMM-based text image modeling and n-gram language modeling [10, 11].

Due to its sequential book structure, the very basic task on RODRIGO is to transcribe it line by line, from the beginning to the end. We assume that an automatic transcription system is used, and that each (automatically) transcribed line is supervised and, if necessary, amended by an expert. Clearly, after processing a block of lines or pages, all supervised transcriptions may be very well used for better (re-)training of image and language models, and thus improving system accuracy.

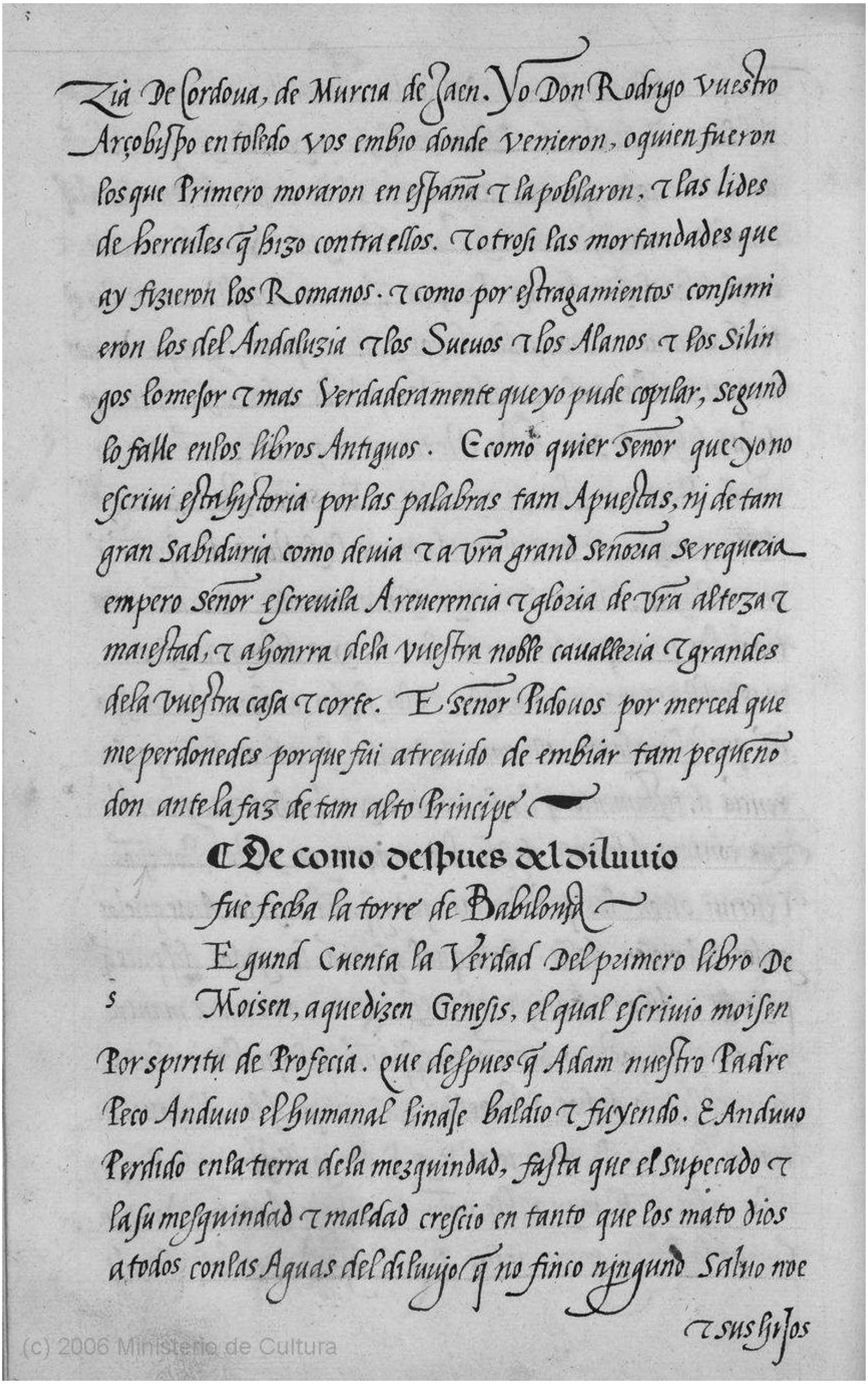
Taking into account the above discussion, we divided RODRIGO into 20 consecutive blocks of 1000 lines each ($1 - 1000$, $1001 - 2000$, ..., $19000 - 20356$). Then, from block 1 to block 19, the system was (re-)trained using all preceding blocks, with block 2 also used for further adjustment of a few, key parameters. After each retraining, the system accuracy was measured in terms of Word Error Rate (WER) on the last block, and the resulting curve is shown in Fig. 2.2, together with a curve showing the part of WER due to the occurrence of out-of-vocabulary (OOV) words.

As expected, the WER decreases as the amount of training data increases. In particular, the system achieves around 37% of WER for the last two blocks, which is not too bad for effective computer-assisted transcription. Although we think that there is room for significant improvements, it must be noted that many errors are caused by the occurrence of out-of-vocabulary words. Note that, most of the system improvement is due to the reduction of out-of-vocabulary words.

2.3 Conclusions

Two new handwritten text databases, GERMANA and RODRIGO, have been presented to facilitate empirical comparison of different approaches to text line extraction and off-line handwriting recognition. On the one hand, GERMANA is the first publicly available database for handwriting research, mostly written in Spanish. On the other hand RODRIGO is completely written in old Castilian (Spanish) by a single author and comparable in size to standard databases. Some preliminary empirical results have been also reported, using standard techniques and tools for preprocessing, feature extraction, HMM-based image modeling, and language modeling. Although we think that there is room for significant improvements, the word error rates obtained are already acceptable for effective computer-assisted transcription.

GERMANA has been published at ICDAR [10], while RODRIGO has been submitted to the LREC conference [12].



zia de Cordoua, de Murcia de Jaen. Yo Don Rodrigo vueſtro
Arçobiſpo en toledo vos embio donde venieron, o quien fueron
losque Primero moraron en eſpaña ⁊ la poblaron, ⁊ las lides
de hercules q̃ hizo contra ellos. ⁊ otroſi las mortandades que
ay fizieron los Romanos. ⁊ como por eſtragamientos conſumi
eron los del Andaluzia ⁊ los Sueuos ⁊ los Alanos ⁊ los Silin
gos lo meſor ⁊ mas Verdaderamente que yo pude copilar, segund
lo falle enlos libros Antiguos. E como quier señor que yo no
eſcriui eſta hiſtoria por las palabras tam Apueſtas, nj de tam
gran Sabiduria como deuia ⁊ a vr̃a grand señoria se requeria
empero señor eſcreuila A reuerencia ⁊ gloria de vr̃a alteza ⁊
maieſtad, ⁊ a honrra dela vueſtra noble caualleria ⁊ grandes
dela vueſtra caſa ⁊ corte. E señor Pidouos por merced que
me perdonedes porque fui atreuido de embiar tam pequeño
don ante la faz de tam alto Principe

De como deſpues del diluuio fue fecha la torre de Babilonia

Egund Cuenta la Verdad del primero libro De
s Moisen, a que dizen Geneſis, el qual eſcriuio moiſen
Por spiritu de Profecia. que deſpues q̃ Adam nueſtro Padre
Peco Anduuo el humanal linaſe baldio ⁊ fuyendo. E Anduuo
Perdido enla tierra dela mezquindad, faſta que el su pecado ⁊
la su meſquindad ⁊ maldad creſcio en tanto que los mato dios
a todos conlas Aguas del diluujo q̃ no finco njngund Saluo noe
⁊ sus hijos

Figure 2.1: Page 15 of RODRIGO.

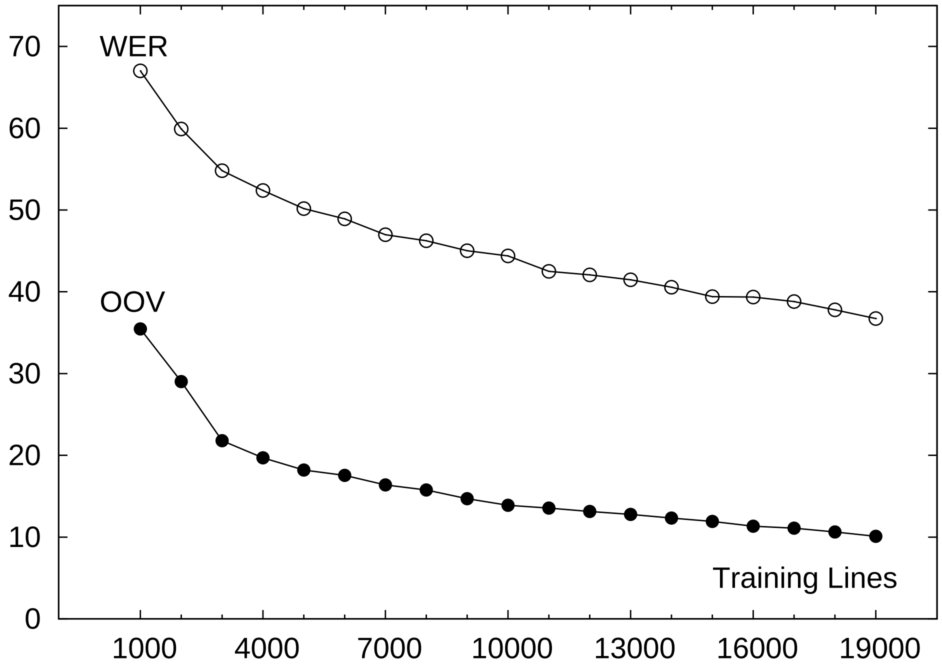


Figure 2.2: Transcription Word Error Rate (WER) on RODRIGO as a function of the block of lines transcribed (line). For each block, the transcription system is trained with all the lines in preceding blocks. Also shown is the part of the WER due to the occurrence of out-of-vocabulary (OOV) words.

CHAPTER 3

The GIDOC prototype

This chapter presents GIDOC (Gimp-based Interactive transcription of old text DOCuments), a prototype designed to work with (large) collections of homogeneous documents, that is, of similar structure and writing styles. GIDOC detects the text block layout and its corresponding lines, offers an intuitive and friendly interface to annotate and recognize transcriptions and uses standard utilities to train the models.

3.1 System Overview

As indicated by its name, GIDOC has been implemented on top of the well-known GNU Image Manipulation Program (GIMP). As GIMP, GIDOC is licensed under the GNU General Public License, and it can be freely downloaded from [13]. To run GIDOC, we must first run GIMP and open a document image. GIMP will come up with its high-end user interface, which is often configured to only show the main toolbox (with docked dialogs) and an image window. GIDOC can be accessed from the menubar of the image window (see Figure 3.1).

As shown in Figure 3.1, the GIDOC includes six entries: *Advanced options, 0: Preferences, 1: Block Detection, 2: Line Detection, 3: HTK Training,* and *4: Transcription. Advanced options* is a second-level menu where experimental features are grouped. *Preferences* opens a dialog to configure global options, as well as more specific options for preprocessing, training and recognition. Some of them are discussed below together with menu entries after *Preferences.*

3.2 Block Detection

Block detection refers to the task of detecting handwritten text blocks in (old) documents. We have worked in this task, with particular emphasis on document collections of homogeneous structure, as in GERMANA and RODRIGO.

Conventional methods for block detection only consider information from the current document image after applying feature extraction methods on it [14]. They ignore the "history" of blocks detected in pages previously seen. In our case, however, the *Block Detection* entry in the GIDOC menu uses a novel text block detection method in which conventional, memoryless techniques are improved with a "history" model of text block positions. Please refer to [15] for details.

3.3 Line Detection

Given a textual block, the *Line Detection* entry in the GIDOC menu detects all its text baselines, which are marked as straight paths. The result can be clearly observed in the example of Figure 3.1. Although each baseline has handlers to graphically correct its position, it is worth noting that the baseline detection method implemented works quite well, at least in pages like that of the example. It is a rather standard projection-based method [14]. First, horizontally-averaged pixel values or black/white transitions are projected vertically. Then, the resulting vertical histogram is smoothed and analyzed so as to locate baselines accurately. Two preprocessing options are included in *Preferences*, first, to decide on the histogram type (pixel values or black/white transitions), and second, to define the maximum number of baselines to be found. When the number of lines detected falls under the maximum defined, GIDOC estimates the mean width between lines and fills gaps broader than it with lines. This method simple correction may detect short lines, such as initial or ending lines of a paragraph.

3.4 HTK Training

GIDOC is based on standard techniques and tools for handwritten text preprocessing and feature extraction, HMM-based image modeling, and language modeling [16]. Handwritten text preprocessing applies image denoising, deslanting and vertical size normalization to a given text (line) image. It can be configured through preprocessing options in *Preferences.* There is an option to use instead a customized procedure, and two options to define (bounds for) the locations of the upper and lower lines, with respect to the baseline.

Feature extraction for HMM modeling consists in transforming the preprocessed image into a *sequence of (fixed-dimension) feature vectors.* There are two, well-known feature extraction methods available in GIDOC. The default method first divides the preprocessed image into a grid of square cells whose size is a small fraction of the image height (e.g. 1/20). Then, each cell its characterized by its normalized gray level and, optionally, by its vertical and horizontal gray-level derivatives. Refer to [16] for more details. The alternative method moves a single-column window left-to-right over the image, and extracts 9 geometrical features at each position [17].

HMM image modeling is carried out with the well-known and freely available *Hidden Markov Model Toolkit (HTK).* [18]. Similarly, language modeling is implemented through the open source *SRI Language Modeling Toolkit (SRILM)* [19]. Both toolkits should be made available to GIDOC for the *HTK Training* entry in the GIDOC menu to properly work.

HTK Training reads the directory of task document images and, for each image, it extracts all its transcribed text lines, if any, together with their corresponding line images. Transcriptions are first preprocessed to isolate special characters (mainly punctuation signs) and expand abbreviations (e.g. *S.M.* is expanded to *Su Magestad*). Using these abbreviations, the lexicon is reduced and an n-gram language model is built from preprocessed transcriptions using a SRILM command which, by default, generates a bigram language model with Knesser-Ney discounting. On the other hand, extracted line images are preprocessed and transformed into sequences of feature vectors so as to train, using their corresponding transcriptions and HTK, continuous density (Gaussian) left-to-right HMMs at character level.

3.5 Transcription

The *Transcription* entry in the GIDOC menu opens the GIDOC interactive transcription dialog (see Figure 3.1). It consists of two main sections: the image section, in the middle part, and the transcription section, in the bottom part. A number of text line images are displayed in the image section together with their transcriptions, if available, in separate editable text boxes within the transcription section. The *current* line to be transcribed or simply supervised is selected by placing the edit cursor in the appropriate editable box. Its corresponding baseline is emphasized (in blue color) and, whenever possible, GIDOC shifts line images and their transcriptions so as to display the current line in the central part of both the image and transcription sections. It is assumed that the user transcribes or supervises text lines, from top to bottom (or in any order desired), by entering text and moving the edit cursor with the arrow keys or the mouse.

Each editable text box has a button attached to its left, which is labeled with its corresponding line number. By clicking on it, its associated line image is extracted, preprocessed, transformed into a sequence of feature vectors, and Viterbi-decoded using HTK and the models trained with *HTK training.* In this way, it is not needed to enter the complete transcription of the current line, but hopefully only minor corrections to the decoded output. Clearly, this is only possible if, first, text lines are correctly detected and, second, the HMM and language models are adequately trained, from a sufficiently large amount of training data. Therefore, it is assumed that transcription is carried out manually in early stages of a transcription task, and then is assisted as described here.

3.6 Experiments

During its development, GIDOC has been used by a paleography expert to annotate blocks, text lines and transcriptions on a new dataset called GERMANA [10]. GERMANA is the result of digitizing and annotating a 764-page Spanish manuscript from 1891, in which most pages only contain nearly calligraphed text written on ruled sheets of well-separated lines. The example shown in Figure 3.1 corresponds to the page 144. GERMANA is solely written in Spanish up to page 180; then, the manuscript includes many parts that are written in languages different from Spanish, namely Catalan, French and Latin.

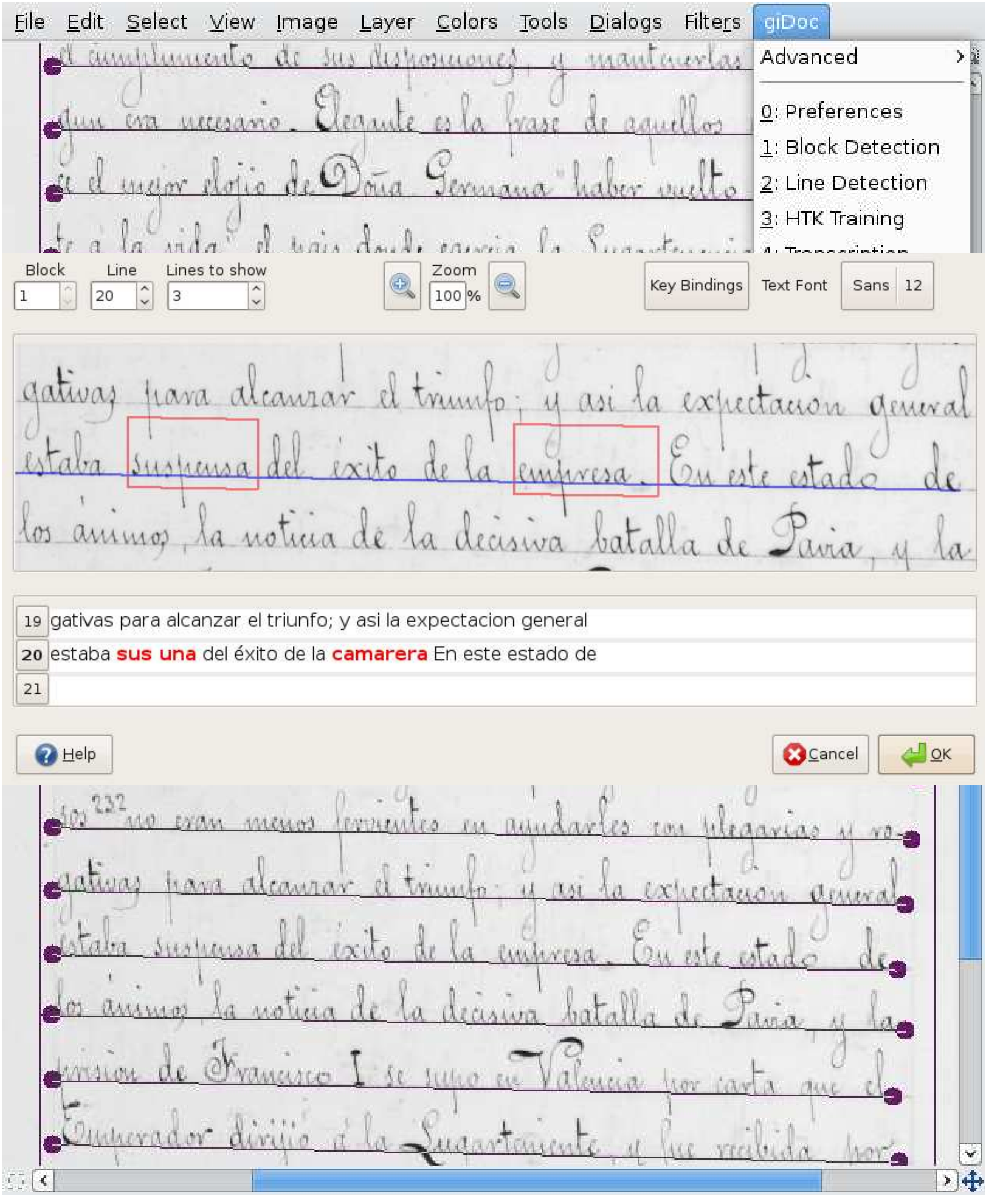
Due to its sequential book structure, the very basic task on GERMANA is to transcribe it from the beginning to the end, though here we only consider its transcription up to page 180. Starting from page 3, we divided GERMANA into 9 consecutive blocks of 20 pages each (18 in block 9) and, on average, 417 lines and 4687 running words. Then, from block 2 (pages 23–42) to block 9 (pages 163–180), each block was automatically transcribed by GIDOC trained with all preceding blocks. The results are plotted in Figure 3.2, in terms of transcription Word Error Rate (WER). To avoid fluctuations due to varying test set complexity, the WER was also computed for a fixed block (block 9) after each GIDOC re-training, and the resulting WER curve has been added to Figure 3.2. Also shown is the part of the WER due to the occurrence of out-of-vocabulary (OOV) words.

As expected, the WER decreases as the amount of training data increases. In particular, GIDOC achieves around 34% of WER for the last two blocks, which is not too bad for effective computer-assisted transcription. The WER curve for block 9 does not differ significantly from that for the next block, though it appears that block 9 is a bit more complicated that all but one (block 7) of its preceding blocks. Regarding the OOV curves, it becomes clear that a considerable fraction of transcription errors is due to the occurrence of unseen words. More precisely, unseen words account for approximately 50% of transcription errors.

3.7 Conclusions

A computer-assisted transcription prototype called GIDOC has been presented for handwritten text in old documents. GIDOC is a first attempt to provide integrated support for interactive-predictive page layout analysis, text line detection and handwritten text transcription. It is build on top of GIMP, and uses standard techniques and tools for handwritten text preprocessing and feature extraction, HMM-based image modeling, and language modeling. As GIMP, GIDOC is licensed under the GNU General Public License, and it can be freely downloaded from Internet. The effectiveness of GIDOC has been empirically demonstrated on the GERMANA database, which is also publicly available on Internet.

The research described in this Chapter has been submitted to the WEBIST 2010 [11] and CHI 2010 [20] conferences. Also, our novel block detection method has been accepted in the DRR 2010 conference [15].


File Edit Select View Image Layer Colors Tools Dialogs Filters giDoc
Advanced
0: Preferences
1: Block Detection
2: Line Detection
3: HTK Training
Block 1
Line 20
Lines to show 3
Zoom 100 %
Key Bindings
Text Font
Sans 12
gativas para alcanzar el triunfo; y asi la expectacion general
estaba suspensa del éxito de la empresa. En este estado de
los ánimos, la noticia de la decisiva batalla de Pavia, y la
Help
Cancel
OK

19 gativas para alcanzar el triunfo; y asi la expectacion general

20 estaba **sus una** del éxito de la **camarera** En este estado de

21

Figure 3.1: Interactive transcription dialog over an image window showing GIDOC menu.

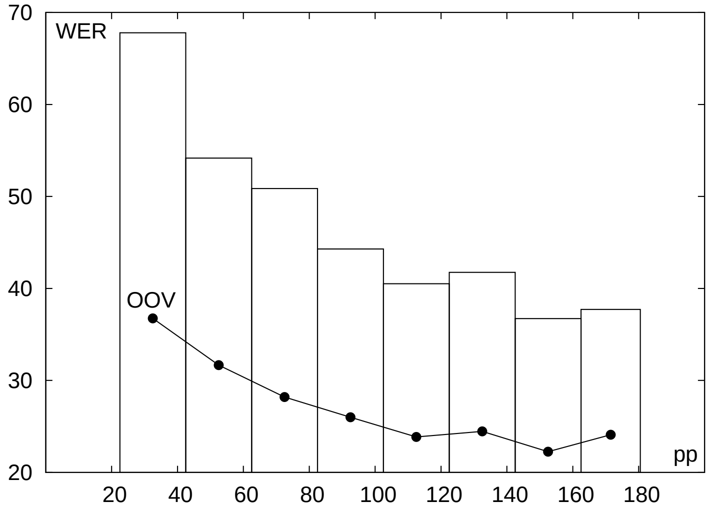


Figure 3.2: Transcription Word Error Rate (WER) on GERMANA as a function of the pages already supervised and thus available for training (training pages). The WER is computed for both, the next 20 pages to supervise (solid line with black circles), and a fixed set comprising pages 163-180 (solid line with white circles). Also shown is the part of the WER due to the occurrence of out-of-vocabulary (OOV) words (dashed lines).

CHAPTER 4

Adaptation from Partially Supervised Transcriptions

Successively produced transcriptions can be used to better adapt image and language models to the task by, for instance, re-training them from the previous and newly acquired transcribed data. However, if transcriptions are only partially supervised, then (hopefully minor) recognition errors may go unnoticed to the user and have a negative effect on model adaptation. In this chapter, we study this effect as a function of the degree of supervision, on two real handwriting transcription tasks of considerable complexity. We also consider three adaptation (re-training) strategies: from all data, only from high-confidence parts, and only from supervised parts. Re-training from high-confidence parts is inspired in the work of Wessel and Ney [21], in which confidence measures were successfully used to restrict unsupervised learning of acoustic models for large vocabulary continuous speech recognition. In this work, however, high-confidence parts include both, unsupervised words above certain confidence threshold, and supervised words. Also, they are used to re-train both, HMMs and the n-gram language model. On the other hand, in order to simulate user actions at different degrees of supervision, we propose a simple yet realistic user interaction model.

4.1 Confidence Measures

Given a classification task, after some recognition have been performed, its uncertainty measure can be used to detect possible errors. In our case, the system could accompany each recognized word, with a measure weighting how certain is of its classification. Then we could use some postprocessing system to re-classify this word, or simply warn the user against relying on it. The uncertainty measure inverse is also known as "confidence measure", which means how certain are your decisions.

In this section we briefly explain the estimation of word-level confidence measures. Taking advantage of the use of standard speech technology by GIDOC, we have adopted a method that has been proved to be very useful for confidence estimation in speech recognition. This method was proposed in [22] and uses posterior word probabilities computed from word graphs as confidence measures.

We define G as a directed, acyclic, weighted graph. The nodes correspond to discrete points in space; a particular frame in the image in our transcription framework. The edges are triplets $[w, s, e]$, where w is the hypothesized word from node s to node e, weighted by the recognition score. All paths between the initial and ending nodes forms a hypothesis $\boldsymbol{f}_1^J$.

The posterior probability of a specific edge (word hypothesis) $[w, s, e]$, given the observations $\boldsymbol{x}_1^T$; can be computed by summing up the posterior probabilities of all paths between the start and ending nodes, containing the edge $[w, s, e]$:

$$P([w,s,e] \mid \boldsymbol{x}_1^T) = \frac{1}{P(\boldsymbol{x}_1^T)} \sum_{\substack{\boldsymbol{f}_1^J \in G: \\ \exists [w',s',e']: \\ w'=w,\, s'=s,\, e'=e}} P(\boldsymbol{f}_1^J,\, [w,s,e],\, \boldsymbol{x}_1^T) \tag{4.1}$$

The probability of the sequence of observations $P(\boldsymbol{x}_1^T)$ can be computed by summing up the posterior probabilities of all word graph hypothesis:

$$P(\boldsymbol{x}_1^T) = \sum_{\boldsymbol{f}_1^J \in G} P(\boldsymbol{f}_1^J,\, \boldsymbol{x}_1^T)$$

The posterior probability defined in Eq. 4.1 does not perform well because a word w can occur in slightly different starting and ending points. This effect is represented in the word graph by different word edges and the posterior probability mass of the word is scattered among the different word segmentations (see Fig. 4.1).

To deal with this problem, we have considered the solution proposed in [22]. Given a specific word (edge) $[w, s, e]$ and a specific point in time $t \in [s, e]$ (time in speech, image position in transcription), we compute the posterior probability of the word w at time t by summing up the posterior probabilities of the word graph edges $[w, s', e']$ with identical word w and for which t is within the interval time $[s', e']$:

$$P_t([w,s,e] \mid \boldsymbol{x}_1^T) = \sum_{t \in [s',e']} P([w,s',e'] \mid \boldsymbol{x}_1^T) \tag{4.2}$$

Based on Eq. 4.2 a better estimate (in practice) consist in fixing the posterior probability for a specific word $[w, s, e]$, as the maximum posterior probability of w in any moment between s and e:

$$P([w,s,e] \mid \boldsymbol{x}_1^T) = \max_{s \le t \le e} P_t([w,s,e] \mid \boldsymbol{x}_1^T) \tag{4.3}$$

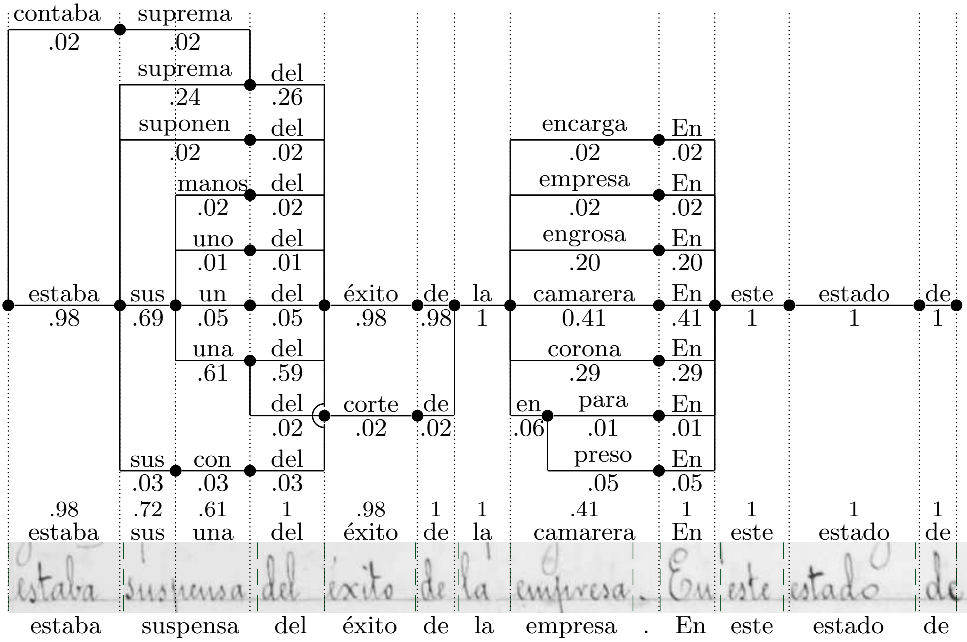


Figure 4.1: Word graph example aligned with its corresponding text line image and its recognised and true transcriptions. Each recognised word is labelled (above) with its associated confidence measure using Eq. 4.1.

The probability computed on Eq. 4.3 is in the interval $[0, 1]$ since, by definition, the sum of the word posterior probabilities for a specific point in time must sum to one.

Apart from the confidence measures presented, other parameters from the word graph can be used to define new ones. We could weight the word posterior probability with its duration, weight the acoustic and language model probabilities, maximize the probability of only one of them, etc. Nevertheless, several combinations have been tested experimentally and the Eq. 4.3 outperformed all other. From here, all confidence measures used in the experiments are calculated using Eq. 4.3.

4.2 User Interaction Model

As said in the introduction, in this chapter we propose a simple yet realistic user interaction model to simulate user actions at different degrees of supervision. The degree of supervision is modelled as the (maximum) number of recognised words (per line) that are supervised: 0 (unsupervised), $1, \ldots, \infty$ (fully supervised). It is assumed that recognised words are supervised in non-decreasing order of confidence.

In order to predict the user actions associated with each word supervision, we first compute a minimum edit (Levenshtein) distance path between the recognised and true transcriptions of a given text line. For instance, the example text line image in Fig. 4.1 is also used in Fig. 4.2 to show an example of minimum edit distance path between its recognised and true transcriptions. As usual, three elementary editing operations are considered: substitution (of a recognised word by a different word), deletion (of a recognised word) and insertion (of a missing word in the recognised transcription). Substitutions and deletions are directly assigned to their corresponding recognised words.

In Fig. 4.2, for instance, there is a substitution assigned to "sus", a deletion assigned to "una", and a second substitution that corresponds to "camarera". Insertions, however, have not direct assignments to recognised words and, hence, it is not straightforward to predict when they are carried out by the user. To this end, we first compute the Viterbi segmentations of the text line image from the true and recognised transcriptions. Given a word to be inserted, it is assigned to the recognised word whose Viterbi segment covers most part of its true Viterbi segment. For instance, in Fig. 4.2, the period, ".", has a true Viterbi segment completely covered by that of the recognised word "camarera", and thus the insertion of "." is assumed to be done when "camarera" is supervised. Note that insertions are assigned to the words being supervise, meaning that an insertion (or several) can be assigned to "sustitution" or "delete" operations.

Figure 4.3 depicts the resulting sentence after having supervised $N = \{1, 2, 3, 4\}$ words in the example shown in Fig 4.2. Although supervision does not result in any editing operation (see 4.3), we assume that a user operation (and its cost) has been performed. Even though our system commits errors, supervising a correct word is the "worst" error our system can commit. Further experiments should take into account these (hard) errors.

In the ideal setup, our confidence measures would detect uncorrectly recognized words. Setting the number of words to be supervised, to the current system errors, should correct almost all system errors. However, our confidence measures are far from perfect, and the number of words to be supervised to perfectly correct the recognized line is greater than the current system errors.

4.3 Adaptation Techniques

In this section, we describe the adaptation techniques used in our work. After supervision of recognized transcriptions, they are used to further improve the system. The systems is re-trained from scratch when a new set of samples is available. The purpose of our techniques is to select the set of samples that will be used in the new re-training. As can be seen in Fig. 4.4, we propose three different re-train techniques:

- From all data re-training (unsupervised) consist in using all available data to train the models. Recognized and not supervised contains errors that suryle will degradate our models.

- From supervised parts re-training, we use to train the models only words that have supervised by the user. Using these approach avoid using errors in training but we do not take benefit from correcly recognized words.

- From high-confidence parts (as in [21]), this approach uses as training samples; both words supervised by the user, and recognized words above some confidence measure threshold. Basically, using this technique we are trying to take the benefits of the two previous techniques.

Most of improvement, when using these adaptation techniques is due to the language model better estimation. HMM character mixture models have shown to have a good tolerance against errors in training, however, language models easily degradate with them. We think that this effect is caused by the estimation of these models. Language models are estimated directly, concretely counting events. On the other hand, HMM models use multiple mixture components trained with the EM algorithm, surely modeling some of the erroneous (noisy) samples in different components.

4.4 Experiments

During its development, GIDOC has been used by a palaeography expert to annotate blocks, text lines and transcriptions on a new dataset called GERMANA [10]. GERMANA is the result of digitising and annotating a 764-page Spanish manuscript from 1891, in which most pages only contain nearly calligraphed text written on ruled sheets of well-separated lines. The example shown in Fig. 4.1 corresponds to the page 144. GERMANA is solely written in Spanish up to page 180; then, the manuscript includes many parts that are written in languages different from Spanish, namely Catalan, French and Latin.

Due to its sequential book structure, the very basic task on GERMANA is to transcribe it from the beginning to the end, though here we only consider its transcription up to page 180. Starting from page 3, we divided GERMANA into 9 consecutive blocks of 20 pages each (18 in block 9). The first two blocks (pp. 3-42) were used to train initial image and language models from fully supervised transcriptions, and optimize some model and recognition parameters. Then, from block 3 to 8, each new block was recognised, partially supervised and added to the training set built from its preceding blocks. We considered three degrees of supervision: 0 (unsupervised), 1 and 3 supervised words per line. Also, as indicated in the introduction, we considered three adaptation (re-training) strategies: from all data, only from high-confidence parts, and only from supervised parts.

From the results in Fig. 4.5, it becomes clear that baseline models can be improved by adaptation from partially supervised transcriptions, though a certain degree of supervision is required to obtain significant improvements. In particular, supervision of 3 words per line leads to a reduction of more than a 10% of WER with respect to unsupervised learning (baseline models), though there is still room for improvement since full supervision (not plotted in Fig. 4.5) achieves a further reduction of 5% (34%). We think that this reduction is mainly due to the language model improvement. At the experiment beginning, language model is trained with few samples, which most are singletons. Increasing the number of bigrams and words, improves the language model estimation and reduces the number of OOV (out-of-vocabulary) words, respectively. As was shown in [10], OOVs reduction is highly correlated with the system improvement. The adaptation strategy, on the other hand, has a relatively minor effect on the results. Nevertheless, it seems better not to re-train from all data, but only from high-confidence parts, or just simply from supervised parts.

Apart from the above experiment on GERMANA, we did a similar experiment on the well-known IAM dataset, using a standard partition into a training, validation and test sets [17]. The training set was further divided into three subsets; the first one was used to train initial models, while the other two were recognised, partially supervised (4 words per line) and added to the training set. The results obtained in terms of test-set WER are: 42.6%, using only the first subset; 42.8%, after adding the second subset; and 42.0%, using also the third subset. In contrast to GERMANA, there is no significant reduction in terms of WER after adding partially supervised data to the training set. We think that this result is due to the more complex nature of the IAM task, as compared with GERMANA, which makes it much more difficult when only a fraction of the training set is available with complete supervision.

On the other hand, we think that IAM experiment is highly influenced by the language model. On the contrary that happened in GERMANA, IAM initial language model is trained using aconsiderable amount of data. Improving these initial model is very difficult, asuming that all lines are perfectly recognized, our training data will only be increased in only 5%. In addition, previous experiments with IAM showed that the most important system part is the language model.

4.5 Conclusions

The adaptation of image and language models from partially supervised data has been studied in the context of computer-assisted handwritten text transcription. A simple yet realistic user interaction model has been proposed to simulate user actions at different degrees of supervision. Empirical results have been reported on two tasks of considerable difficulty. We have shown how the use of confidence measures can help to reduce drastically the supervision effort improving the transcription accuracy.

In sequential frameworks, where non-extern models can be used, such as GERMANA, it has been shown that supervising a few words achieves almost (5% difference) the same result than complete user supervision. This framework would perfectly suit, transcriptions task where a little error is tolarated, e.g. word spotting. Adaption tecniques have proved to further improve system efficiency. However, when external data can be used, user supervision does not significantly improve the system performance.

In the future, we plan to improve our current confidence measures so a non-uniform number of words is corrected per line. Re-training from scratch all models performs well on single-topic tasks, we plan to study the use of the framework presented in multi-topic tasks.

Research refered in this chapter is published in the ICMI-MLMI joint conference [23].

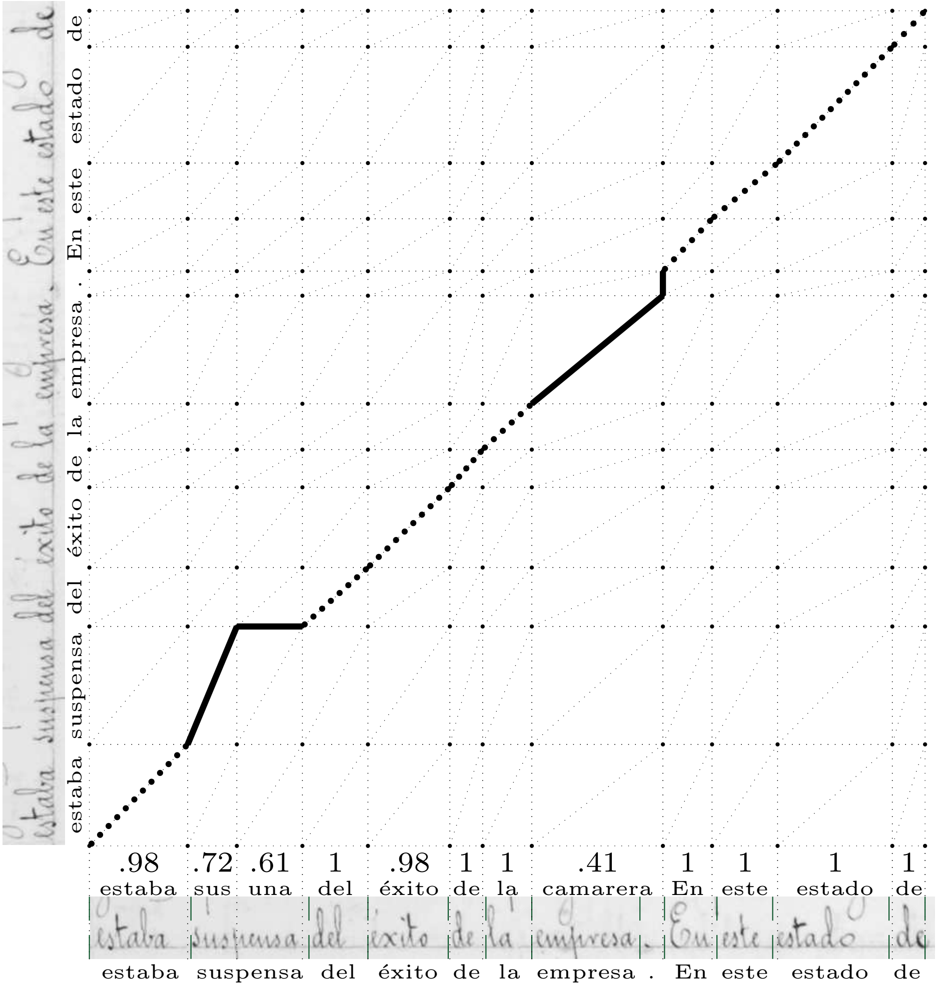


Figure 4.2: Example of minimum edit distance path between the recognised and true transcriptions of a text line image.

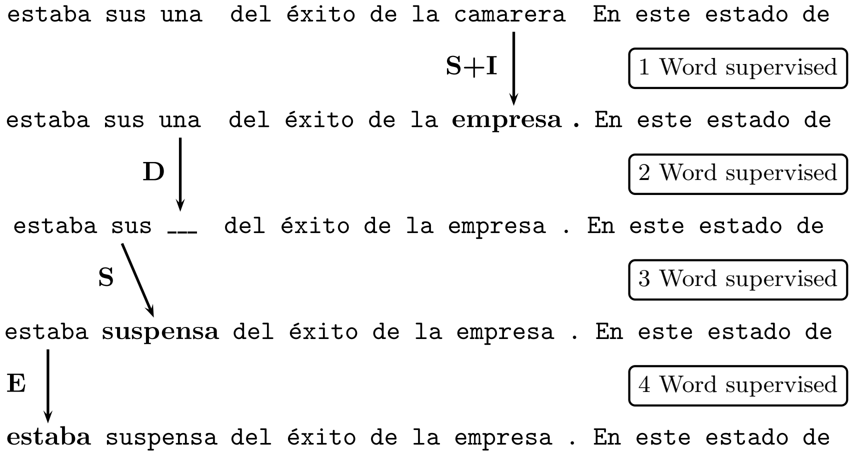

estaba sus una del éxito de la camarera En este estado de
S+I
1 Word supervised
estaba sus una del éxito de la empresa . En este estado de
D
2 Word supervised
estaba sus ___ del éxito de la empresa . En este estado de
S
3 Word supervised
estaba suspensa del éxito de la empresa . En este estado de
E
4 Word supervised
estaba suspensa del éxito de la empresa . En este estado de

Figure 4.3: Example of supervising $N = \{1, 2, 3, 4, 5\}$ recognized words with our user interaction model. Edit operations are marked as: **E**qual, **S**ubstitution, **D**eletion and **I**nsertion.

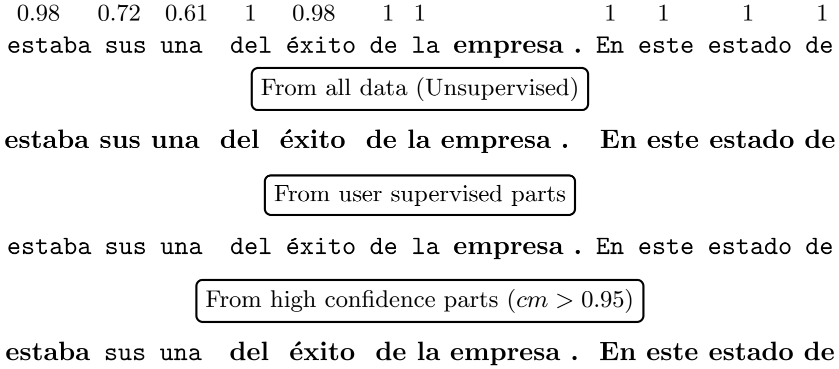

0.98 0.72 0.61 1 0.98 1 1 1 1 1 1
estaba sus una del éxito de la empresa . En este estado de
From all data (Unsupervised)
estaba sus una del éxito de la empresa . En este estado de
From user supervised parts
estaba sus una del éxito de la empresa . En este estado de
From high confidence parts (cm > 0.95)
estaba sus una del éxito de la empresa . En este estado de

Figure 4.4: Words (marked in bold) which will be used in the next retraining, when using the different adaptation technique. The words "empresa ." are fixed because they have been supervised by the user. Recognized word confidence measure is placed above it.

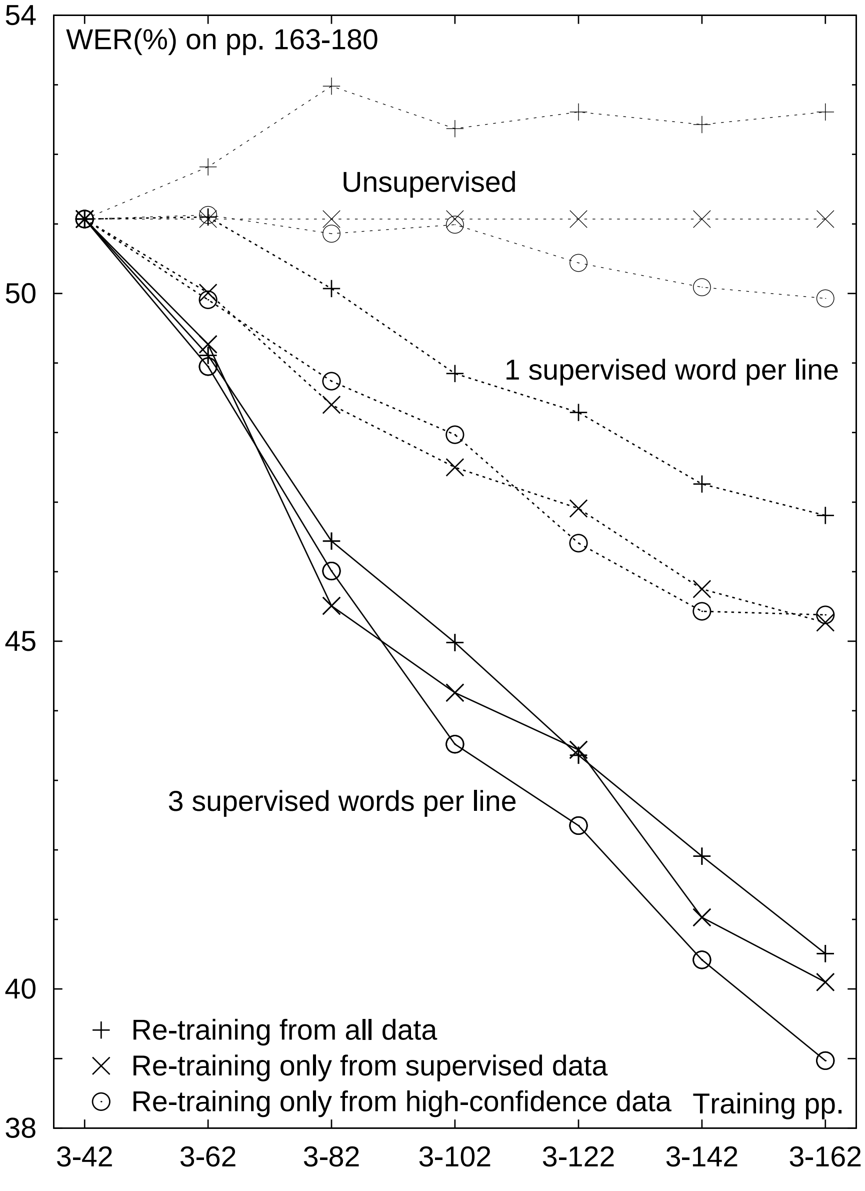


Figure 4.5: Test-set Word Error Rate (WER) on GERMANA as a function of the training set size (in pages), for varying degrees of supervision (supervised words per line).

CHAPTER 5

Balancing Error and Supervision Effort

In this chapter, we study how to automatically balance recognition error and supervision effort. In the previous chapter, we have compared several model adaptation techniques from partially supervised transcriptions. It has been shown that it is better not to adapt models from all data, but only from high-confidence parts, or just simply from supervised parts. More importantly, it has been shown that a certain degree of supervision is required for model adaptation though, it remains unclear how to adjust it properly. In this work, we propose a simple yet effective method to find an optimal balance between recognition error and supervision effort. The user decides on a maximum tolerance threshold for the recognition error (in non-supervised parts), and the system adjusts the required supervision effort on the basis of an estimate for this error.

5.1 Predicting the Error

Annotation of an old text document is a time consuming task. Nowadays, automatic transcriptions of old text documents is far from perfect. Fortunately, when error rates fall around 35%, an automatic system can be used to speed up the annotation process. In previously presented system [11, 6], this kind of systems have been used successfully. Nevertheless, if perfect annotation is desired, it still requires to fully supervise the system output, which increases the user effort. On the other hand, if transcription errors are tolerated, later user supervision will not be required, decreasing the user effort.

When a limited number of errors are tolerated, our main objective should be to decrease the user effort. In a transcription framework, the user effort required to perfectly correct a recognized sentence, is exactly the number of edit operations to transform it to the reference sentence. These edit operations are the cost of transforming the incorrect words in the sentence. Ideally, we need to know which are the incorrect words of a recognized line, so we can fix the error or the user effort. However, we do not have this information. Confidence measures help us to find out incorrect words, but we still need an estimation of the error in the sentence.

In interactive-predictive frameworks, the system asks for user supervision. This information can be used to estimate the current error committed by our system. If next sentences to be recognized are taken from the same source than previous ones, we could expect them to follow the same error estimations. Given these assumptions, we can build a system able to estimate the expected error of a given line. In conclusion, using the user supervisions we can estimate the next sentences error, to then decide if supervisions are necessary.

5.2 Balancing Error and Supervision Effort

Given a collection of reference-recognized transcription pairs, its WER may be simply expressed as:

$$WER = \frac{E}{N}$$

where E is the total number of editing operations required to transform recognized transcriptions into their corresponding references, and N is the total number of reference words. In this work, however, we need to decompose these three variables additively, as

$$\begin{aligned} WER &= WER^{+} + WER^{-} \\ E &= E^{+} + E^{-} \\ N &= N^{+} + N^{-} \end{aligned}$$

where the superscripts $^{+}$ and $^{-}$ denote supervised and unsupervised parts, respectively, and thus

$$WER^{+} = \frac{E^{+}}{N}$$

and

$$WER^{-} = \frac{E^{-}}{N}$$

In order to balance error and supervision effort, we propose the system to ask for supervision effort only when WER^{-} becomes greater than a given, maximum tolerance threshold, say WER^{*}. However, as we do not know the values of E^{-} and N^{-}, they have to be estimated from available data. A reasonable estimate for N^{-} is simply

$$\hat{N}^{-} = \frac{N^{+}}{R^{+}} R^{-}$$

where R^+ and R^- denote the number of recognized words in the supervised and unsupervised parts, respectively. Similarly, a reasonable estimate for E^- is

$$\hat{E}^- = \frac{E^+}{R^+} R^-$$

and thus the desired estimate for WER^- is

$$\widehat{WER}^- = \frac{\frac{E^+}{R^+} R^-}{N^+ + \frac{N^+}{R^+} R^-}$$

Each recognized word will be accepted without supervision if it does not lead to a WER^- estimate greater that WER^*.

Note that the above estimate for WER^- is pessimistic, since it assumes that, on average, correction of unsupervised parts requires similar editing effort to that required for supervised parts. However, the user is asked to supervise recognized words in increasing order of confidence, and hence unsupervised parts should require less correction effort. In order to better estimate WER^-, we may group recognized words by their level of confidence c, from 1 to a certain maximum level C, and compute a c-dependent estimate for E as above,

$$\hat{E}_c^- = \frac{E_c^+}{R_c^+} R_c^-$$

where E_c^+, R_c^+ and R_c^- are c-dependent versions of E^+, R^+ and R^-, respectively. The global estimate for E is obtained by simply summing these c-dependent estimates,

$$\hat{E}^- = \sum_{c=1}^{C} \hat{E}_c^-$$

and, therefore, the estimate for WER^- becomes

$$\widehat{WER}^- = \frac{\sum_{c=1}^{C} \frac{E_c^+}{R_c^+} R_c^-}{N^+ + \frac{N^+}{R^+} R^-}$$

which reduces to the previous, pessimistic estimate when only a single confidence level is considered ($C = 1$).

5.3 Experiments

During its development, GIDOC has been used by a paleography expert to annotate blocks, text lines and transcriptions on a new dataset called GERMANA [10]. GERMANA is the result of digitizing and annotating a 764-page Spanish manuscript from 1891, in which most pages only contain nearly calligraphed text written on ruled sheets of well-separated lines. GERMANA is solely written in Spanish up to page

180; then, the manuscript includes many parts that are written in languages different from Spanish, namely Catalan, French and Latin.

Due to its sequential book structure, the very basic task on GERMANA is to transcribe it from the beginning to the end, though here we only consider its transcription up to line 3700 (page 177). For this part, we consider its transcription under three tolerance thresholds on the recognition error (in unsupervised parts): 0% (fully supervised), 9% (one recognition error per line, on average), and 18%.

We divided GERMANA into consecutive blocks of 100 lines each (37 blocks). The first two blocks were used to train initial image and language models from fully supervised transcriptions. Then, from block 3 to 37, each new block was recognized, partially supervised as discussed in the preceding section for $C = 4$ confidence levels, and added to the previous training set. The first three confidence levels correspond, respectively, to the first three words in each line that were recognized with smaller confidence; the remaining recognized words were all grouped into the fourth confidence level. All these levels are initialized so the first WER estimation is 100%. Re-training of image and language models was carried out from only high confidence parts (details in Sec. 4.3). On the other hand, simulation of user supervision actions on each recognized word was done in accordance with the user interaction model described in Sec. 4.2. The results are shown in Fig. 5.1 in terms of WER on transcribed lines (excluding the first 200).

From the results in Fig. 5.1, it becomes clear that the proposed balancing method takes full advantage of the allowed tolerance to reduce supervision effort. Moreover, the total WER of the system trained with partial transcriptions does not deviate significantly from that of the fully supervised system. The average user effort reduction ranges from 17% (for $WER^* = 9\%$) to 33% (for $WER^* = 18\%$). That is, if one recognition error per line is allowed on average ($WER^* = 9\%$), then the user will save a 17% of the supervision actions that are required in the case of a fully supervised system. Here, supervision actions refers to elementary editing operations, and also to check that a correctly recognized word is certainly correct.

The results presented in Fig. 5.1 are quite satisfactory, we have observed that the proposed balancing method does clearly favor supervision of low confidence words over those recognized with high confidence. In terms of user effort, at the end of 9, 18 and 27 experiments, the user have performed 70.78%, 51.33% and 38.20% respectively of editions operations required.

On the other hand, in the 9 experiment, around 35% of edition operations in the not supervised part are errors; in the 18 and 27 experiments this percentage is 35 and 40 respectively. This means that even that most of not supervised words are correct, there is still great room for further improvement. We think that this behavior can be alleviated by using more confidence levels or, using a better estimate of the reference words number, or using another kind of confidence measures.

5.4 Conclusions

A simple yet effective method has been proposed to find an optimal balance between recognition error and supervision effort in interactive-predictive handwriting recognition. The user decides on a maximum tolerance threshold for the recognition error (after supervision), and the system adjusts the required supervision effort on the basis of an estimate for this error. Empirical results have been reported showing the effectiveness of the proposed method. Current work is underway to develop improved variants of this method, and to obtain more empirical results in transcription tasks other than GERMANA.

Balancing the error can be used in a wide area of applications. This system can obtain an initial transcription (with little user effort), which perfectly fits word spotting applications, where annotation errors can be tolerated.

As future work, we plan to extend this technique to other application field, such as speech recognition or machine translation. In tasks that implies multiple languages or topics, our work could be improve using multiple error estimation systems. Another important aspect would be, to test the possibility of changing the maximum WER threshold during the transcription process.

The research presented in this chapter has been accepted in the international conference IUI [24].

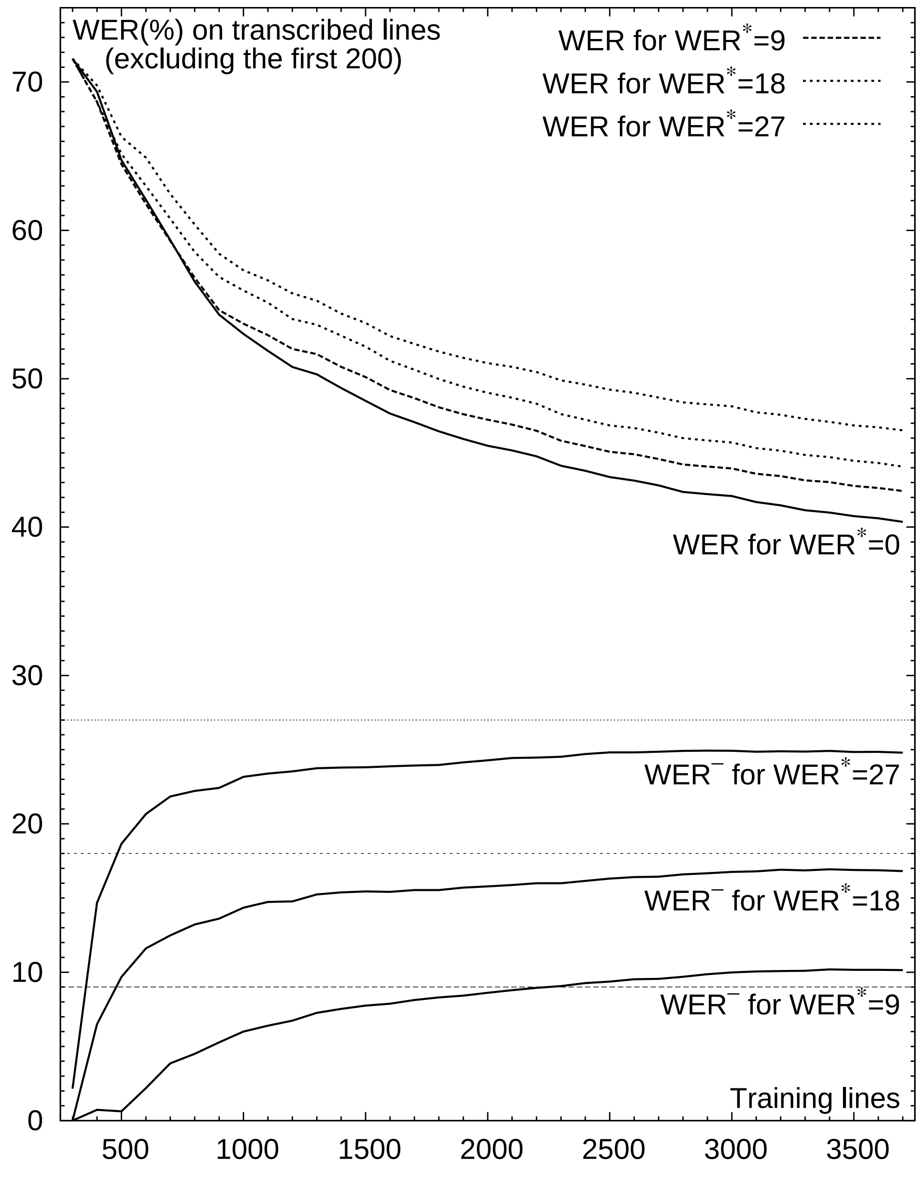


Figure 5.1: Word Error Rate (WER) on transcribed lines (excluding the first 200), as a function of the (number of) training lines, for varying tolerance thresholds on the recognition error (in unsupervised parts).

CHAPTER 6

Kernel Regression in Machine Translation

In this chapter we present a new approach to machine translation. In this approach, both source and target strings are mapped to natural feature vectors. Then a *translation mapping* is learnt between feature vector domains. Given a source string $\boldsymbol{x}$, the translation process consists in mapping it to a feature vector $\boldsymbol{u}$ and then, mapping this vector $\boldsymbol{u}$ to its corresponding target feature vector $\boldsymbol{v}$. The latter mapping, the so-called translation mapping is learnt by regression techniques. Finally, the pre-image set for the target feature vector $\boldsymbol{v}$ must be found. This problem is referred as the "Pre-image" problem. The focus of this chapter is to solve this problem in a regression-based machine translation framework.

6.1 Introduction

Some previous works such as [25], have explore the idea of learning the translation as a regression problem. These works do not handle the pre-image problem directly but use the model as a score to the standard statistical machine translation systems. Specifically, they use a phrased-based statistical machine translation model [26] and use the kernel regression model as score to the phrased-based search. This approach does not make the best of the regression approach, as proposed in [27], losing some of its main advantages.

On the contrary, the pre-image search proposed in [27] is adapted in this work to the peculiarities in the machine translation problem. This aim is achieved by building the DeBruijn Graph [28] for a target feature vector and then finding eulerian paths within it. However, due to the high dimensionality of feature vectors, problems arise.

6.2 The Training Process

The aim of machine translation is to learn a mapping from a source language $X^\star$ to a target language $Y^\star$, i.e. $f : X^\star \rightarrow Y^\star$, where X is the source vocabulary and Y is the target vocabulary. In statistical machine translation the optimal translation function f is designed based on Bayes' decision theory [29]

$$f(\boldsymbol{x}) = \underset{\boldsymbol{y} \in Y^\star}{\operatorname{argmax}} \, p(\boldsymbol{y}|\boldsymbol{x}) \tag{6.1}$$

where $p(\boldsymbol{y}|\boldsymbol{x})$ is approximated by

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(C(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in Y^\star} \exp(C(\boldsymbol{x}, \boldsymbol{y}'))} \tag{6.2}$$

since the exponential function is monotonically increasing, Eq. (6.1) simplifies to

$$f(\boldsymbol{x}) = \underset{\boldsymbol{y} \in Y^\star}{\operatorname{argmax}} \, C(\boldsymbol{x}, \boldsymbol{y}) \tag{6.3}$$

where $C(\boldsymbol{x}, \boldsymbol{y})$ is a score function, which is modelled by a feature set $\{h_k\}_1^K$ of theoretically and heuristically motivated functions [30]

$$C(\boldsymbol{x}, \boldsymbol{y}) = \sum_k \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y}) \tag{6.4}$$

where λ_k are the model parameters that weight the feature function relevance. In state-of-art translation systems [31], these feature functions are mainly modelled by logarithms of probability models, such as the phrase-based models [26].

However, we propose a method for learning the translation function f based on regression techniques. In order to configure the regression framework, the source strings $\boldsymbol{x} \in X^*$ are mapped to the natural domain $\boldsymbol{u} \in U \subset \mathbb{N}^{D_1}$ via a source mapping function ϕ_X, i.e. $\boldsymbol{u} = \phi_X(x)$. Similarly, the target strings $y \in Y^*$ are mapped to another natural domain, $v \in V \subset \mathbb{N}^{D_2}$, via a target mapping function ϕ_Y, i.e. $\phi_Y(y)$. Although both source and target mappings are not required to be of the same type, we will henceforth assume so. Then, we define the mappings, ϕ_X and ϕ_Y, as the function, ϕ_n, that generates a n-gram count vector from a given string, $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively. More accurately, $\phi_n(x) = \{|x|_{\boldsymbol{u}_1}, \ldots, |x|_{\boldsymbol{u}_n}\}$ with $\boldsymbol{u}_i$ standing for the i-th n-gram in lexicographic order of the vocabulary X, and $|\boldsymbol{x}|_{\boldsymbol{u}}$ is the number of occurrences of $\boldsymbol{u}$ in $\boldsymbol{x}$. For instance, the string $\boldsymbol{x} =$ "*aaabb*" of the language $X^* = \{a, b\}^*$ will correspond to the bigram mapping output $\boldsymbol{u} = \phi_2(\boldsymbol{x}) = (2, 1, 0, 1)$.

The feature vector mapping $u \in U \subset \mathbb{N}^{D_1}$ is useful when comparing strings, since it helps to define a kernel between strings as follows:

$$K_n(\boldsymbol{x}, \boldsymbol{x}') = \phi_n(\boldsymbol{x})\phi_n(\boldsymbol{x}')^T = \sum_{\boldsymbol{u} \in \Sigma^n} |\boldsymbol{x}|_{\boldsymbol{u}} |\boldsymbol{x}'|_{\boldsymbol{u}} \tag{6.5}$$

where $K_n(\boldsymbol{x}, \boldsymbol{x}')$ ranges from the number of common n-grams if both strings are equal, to zero if they totally differ.

$$\begin{tikzcd}
X \arrow[r, "f(\cdot)"] \arrow[d, "\phi_X(\cdot)"'] \arrow[dr, "g(\cdot)"] & Y \arrow[d, bend right, "\phi_Y(\cdot)"'] \\
U \arrow[r, "h(\cdot)"] & V \arrow[u, bend right, "\phi_Y^{-1}(\cdot)"']
\end{tikzcd}$$

Figure 6.1: Machine translation kernel regression scheme

Once the string-to-vector mapping is defined, the training problem is restated as a regression problem where a source-to-target mapping must to be found, i.e. finding the translation mapping $h : U \to V$. Given such a mapping h and two sentences, a source sentence $\boldsymbol{x}$ and its translation $\boldsymbol{y}$; we define a string to target feature mapping, $g : X^\star \to V$, as follows

$$g(\boldsymbol{x}) = h(\phi_X(\boldsymbol{x})) = \phi_Y(\boldsymbol{y}) \tag{6.6}$$

Given a source string $\boldsymbol{x}$, the proposed translation method consists in mapping it to U via $\phi_X(\boldsymbol{x})$ and then, mapping it to V with the translation function $h(\boldsymbol{x})$. Afterwards, given the target feature vector $\boldsymbol{v}$ obtained from the translation mapping, we compute its pre-image set $\phi_Y^{-1}(\boldsymbol{v})$. Figure 6.1 depicts a general scheme of the whole translation process.

6.2.1 The Linear Regression

The function h maps between two natural domains $h : \mathbb{N}^{D_1} \to \mathbb{N}^{D_2}$. Since discrete regression problems yield complicated training algorithms, we learn an extension of this function $\bar{h}$ that approximates the natural domains by real domains, i.e. $\bar{h} : \mathbb{R}^{D_1} \to \mathbb{R}^{D_2}$. We further assume that our regression problem is linear, that is to say, that the mapping function $h(\boldsymbol{u})$, and hereby its extension $\bar{h}$, can be approximated with a linear function, i.e. $\bar{h}(\boldsymbol{u}) = \mathbf{W}\boldsymbol{u}$. Note that in this case the string-to-feature vector mapping g, is simplified to $g(\boldsymbol{x}) = \mathbf{W}\phi_X(\boldsymbol{x})$. Given a set of sentences, $(\boldsymbol{x}_m, \boldsymbol{y}_m)_1^M$, we can learn the optimal $\hat{\mathbf{W}}$ matrix using the regularised least square as follows

$$\hat{\mathbf{W}} = \arg\min_{W} \sum_{m=1}^{M} ||\mathbf{W}\phi_X(\boldsymbol{x}_m) - \phi_Y(\boldsymbol{y}_m)||^2 + \gamma||\mathbf{W}||_F^2 \tag{6.7}$$

where $||\cdot||_F$ refers to the Frobenius norm and where $\gamma > 0$ is the regularization term. The aim of the regularisation term is to avoid the weights matrix $\mathbf{W}$ to contain large weights, which is a clear evidence of overtraining.

The solution to Eq. (6.7) is unique and is found by differentiating the expression and equaling it to zero

$$\hat{\mathbf{W}} = \mathbf{M}_Y(\mathbf{K}_X + \gamma\mathbf{I})^{-1}\mathbf{M}_X^T \tag{6.8}$$

where $\mathbf{M}_Y = [\phi_Y(y_1), \ldots, \phi_Y(y_M)]$ is the $D_2 \times M$ matrix of which j-th column vector is $\phi_Y(y_j)$, where $\mathbf{M}_X = [\phi_X(x_1), ..., \phi_X(x_M)]$ is the analogous for the source strings, and where $\mathbf{K}_X$ is the Gram matrix associated to the kernel K_n applied to the source samples, i.e. $[\mathbf{K}_X]_{ij} = K_n(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for all i, j such that $1 \leq i, j \leq M$.

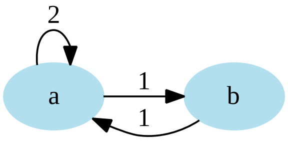


Figure 6.2: A DeBruijn graph for bigrams and the feature vector $\boldsymbol{v} = \{2, 1, 1, 0\}$.

6.3 The decoding Problem

In the proposed framework, the regression output for a given source string $\boldsymbol{x}$, is a feature vector $\boldsymbol{v}$ which represents the occurrence counts of each target n-gram in its respective target translation $\boldsymbol{y}$. Any target string $\boldsymbol{y}$, that produces the same count vector $\boldsymbol{v}$ by means of the target mapping $\phi_Y(\boldsymbol{y})$ is regarded as a possible translation of the source sentence $\boldsymbol{x}$. Therefore, the decoding problem or the pre-image problem is theoretically constrained to a subdomain of Y^*.

The decoding or pre-image problem is stated as the problem of finding which target sentences are represented as given a feature vector, i.e., to find the set $\phi_Y^{-1}(g(\boldsymbol{x}))$. The pre-image of $g(\boldsymbol{x})$ is not unique since each reordering of the same counts leads to the same target feature vector $\boldsymbol{v}$. To further understand the problem we give a simple example. We assume the target language in the example is given by $Y^\star = \{a, b\}^\star$ and that the mapping function is ϕ_2, i.e. counting the number of possible bigrams. This implies that the count vector has four dimensions, one for each of the possible bigrams $\{aa, ab, ba, bb\}$. Since the dimensions are sorted in lexicographic order, the first dimension v_1 represents the occurrences of the bigram aa, the second dimension v_2, the occurrences of ab; and so on. If the regression output for a source sentence $\boldsymbol{x}$ is $\boldsymbol{v} = g(\boldsymbol{x}) = \{2, 1, 1, 0\}$, then its pre-image set is $\phi_Y^{-1}(\boldsymbol{v}) = \{aaaba, aabaa, baaab\}$.

When dealing with natural feature vectors the pre-image problem has a well-known solution [28]. First, it is needed to build the so-called DeBruijn graph as follows: all the $(n-1)$-gram sequences represent a different node in the graph, and edges are added going from a node, $a_1 a_2 \ldots a_{n-1}$, to a node, $a_2 \ldots a_{n-1} a_n$, if they have a positive weight, which is the count of the n-gram $a_1 \ldots a_n$ in $\boldsymbol{v}$. In this graph, each Eulerian path[a] corresponds to a target string in the pre-image set. The DeBruijn graph of the proposed example is shown Fig. 6.2.

At this point several problems arise in practice:

- The translation regresion is real-valued $\bar{h}$ instead of natural-valued h. Eulerian paths are not correctly defined in this case, since the DeBruijn technique cannot be directly applied in real-valued feature vectors.

- There are unknown source and target n-grams, those not appearing at the training samples. This makes the target feature vector $\boldsymbol{v}$ not to define a unique con-

[a] A path inside a graph visiting each edge the number of times indicated by its weight

nected DeBruijn graph, but one with multiple isolated connected components.

- Even if we obtain all the possible eulerian paths within the graph, that represents all the possible translation, there is no way to select the proper translation corresponding to the source string.

6.3.1 The Aliasing Problem

The result obtained from the regression is a real-valued g function where the meaning of each of the target vector dimensions v_k is not clear. For instance, we take the example in which the target vocabulary is given by $Y^\star = \{a, b\}^\star$ and a bigram mapping is used $\phi_2(\cdot)$. In this example, the obtained target feature vector for a given source string could be $\boldsymbol{v} = (1.2, 0.5, 0.8, 0.25)$ instead of the perfect regression $\boldsymbol{v}' = (2, 1, 1, 0)$.

For deeply understanding the implication of any approximation, we must analyse the effect of variations in the natural-valued feature vector. Assuming that the correct regresion is given by $\boldsymbol{v} = (2, 1, 1, 0)$ then the pre-image set is $\phi^{-1}(\boldsymbol{v}) = \{abaa, aaba, baaab\}$. In the case in which the regresion produces $\boldsymbol{v} = (2, 1, 1, 1)$, the pre-image set changes to $\phi^{-1}(\boldsymbol{v}) = \{abbaa, aabba, baaabb, bbaaab\}$. Therefore, adding (or subtracting) 1 to any count dimension incurs in one extra (or less) word.

The search method originally proposed in [27] is to round the real vector and then, build a DeBruijn graph from it and search eulerian paths whittin the graph. However, the best solution takes into account the previously discussed regression errors. For this aim we define the *Levensthein loan*[b] as the real increment or decrement that must be added to a given real vector in order to convert it to the natural domain. The Levensthein loan can be understood as the average Levensthein error of the correct hypothesis with respect to unknown reference.

In order to find the pre-image for real-valued feature vectors, we build a weighted graph in a similar way the DeBruijn graph is built. The edges represent the real count of each n-gram according to the real-valued feature vector, instead of actual natural counts. During the search, the Levensthein loan is used to add or substract any necessary amount to the eulerian path simulating, in this way, a natural vector. In summary, the search algorithm looks for a path that uses the more of the weights in the graph and ask for the lowest possible loan.

6.3.2 The Unknown n-grams Problem

In practice, it is common to find unknown n-grams when mapping strings to the feature vector count domain. This problem is stressed as the n becomes larger. Usually, this leads to DeBruijn graphs with isolated connected components and without eulerian paths covering the full sentence.

A way to amend the problem is to apply "backing-off" to lower order models during the search. That is to say, when searching for a path, the search is simultaneously performed in different l-gram graphs ($1 \leq l \leq n$). Higher values of l are more restrictive during the search and also capture more structural dependencies but more

[b] Named after the Levensthein distance [32]

number of parameters must be estimated in the regression. When the search at the highest order graph cannot continue building an eulerian path, the search backs off one order level with a penalisation factor, similarly to backing-off smoothing in language modeling [33]. This process goes on recursively until the unigram level if needed.

6.3.3 Adding Scores to the Search

Lenvensthein loan score is not enough to select just one target string from the real-valued feature vector. Recall that even in the theoretical situation this would not be possible since several sentences can be built from different reordering of the same n-gram counts. These sentences have the same loan and therefore, we have no way to discriminate among them.

Obviously, this problem is solved by adding an additional score to the target string $\boldsymbol{y}$ and consequently to each path in the pre-image search. Although, several scores can be proposed such as the probability given by statistical translations models, we have adopted in this first work a language model probability, specifically the n-gram language model [33]. In summary, the score in Eq. (6.3) for a given pair $(\boldsymbol{x}, \boldsymbol{y})$ is defined as follows for the proposed kernel regression model

$$C(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{N} \lambda_i \,\mathrm{sc}_i(\boldsymbol{x}, \boldsymbol{y}) \right) + \lambda_{lm} p_n(\boldsymbol{y}) + \lambda_l \exp^{(-|\boldsymbol{y}|\alpha)} \tag{6.9}$$

where sc_i is the Levensthein loan for the target string $\boldsymbol{y}$ at the i-th level graph; $p_n(\boldsymbol{y})$ is the language probability model for the target sentence $\boldsymbol{y}$, and the last term represents a word-bonus score to counteract the language model bias towards short sentences.

6.4 Experimental Results

We have carried out experiments on the categorized EuTrans corpus [34]. This corpus comprises tourist requests at hotel front desks. Categorized EuTrans consists in 13000 pair of sentences divided into three sets: 9000 sentences to train, 1000 sentences were left out for validation and 3000 sentences for testing. There are 420 source and 230 target categories. The corpora perplexity is 3.95 and 4.73 for the target and source language respectively.

Three kernel n-gram models where trained: one built from bigrams consisting on 2145 source bigrams and 929 target bigrams, another built from trigrams consisting on 5048 source trigrams and 2102 target trigrams; and the concatenation of both as described in 6.3.2. We trained increasing sizes of n-gram language models from 2 to 5 estimated by the SRILM toolkit [19]. The score weights, λ, in Eq. (6.9) were adjusted by the Downhill Simplex algorithm [35] for optimizing the BLEU score [36] on the validation set. The proposed system was compared with the Moses [31] baseline system, in which we have limited the phrase length to 7 allowing reordering and optimizing the parameters on a validation set. The Moses baseline system scored 92.3 points of BLEU compared to the 95.5 points of our best system. A practical behaviour analysis of the proposed method is enclosed in the table 6.1. The results on this simple

Table 6.1: Results in terms of BLEU, on categorized EuTrans. N stands for the order of the kernel whilst the LM order row stands for the order of the n-gram language model (0 no LM).

Model \ LM order	0	2	3	4	5
$N = 2$	86.8	93.4	94.1	**95.0**	94.8
$N = 3$	94.0	**94.4**	93.8	94.1	94.3
$N = 3 + (N = 2)$ Backoff	95.3	**95.5**	95.5	95.5	95.4

task are encouraging since almost all results surpass the baseline system. It can be observed that adding language information considerably improves the bigram system. However the trigram system is not benefited from the language model information, probably because of the corpus simplicity. Finally as expected, the search smoothing with "back-off" improves the results.

6.5 Conclusions

Kernel regression models are a new and encouraging approach to the machine translation field. In this work we have proposed a complete regression framework to machine translation by proposing a regression-based search. In contrast, other works perform the search by means of a phrase-based decoding. In addition, we have explored the idea of adding language information to rank the target strings among all the pre-image set. In a future work, different models apart from the n-gram language model will be added to rank the target strings, such as IBM word models and phrased-based models.

Nevertheless several problems arise when complex corpus are used. To deal with them, further optimizations such as perform the model estimation through Cholesky incomplete decomposition or subset selection techniques are left for further research. Other possible improvements in the process would be using different kernel functions for comparing strings and use quadratic regression instead of linear regression as the regression model.

The work presented in this chapter is published in IBPRIA [37], and won the best paper prize award.

CHAPTER 7

Conclusions

This work have contributed to the development of advanced techniques and interfaces for the analysis, transcription and translation of images of old archive documents, following an interactive-predictive approach. More specifically, the contributions described in this work are the following:

GERMANA & RODRIGO: Preparation of databases of old text documents. Annotation of digitized pages from two historical document collections, GERMANA and RODRIGO, have been presented to facilitate empirical comparison of different approaches to text line extraction and off-line handwriting recognition. This work have generated two articles in international conferences:

- **ICDAR-2009:** D. Pérez, L. Tarazón, **N. Serrano**, F. Castro, O. Ramos and A. Juan. The GERMANA database. *Proceedings of the 10th ICDAR.* Barcelona (Spain). July 2009.
- **LREC-2010: N. Serrano**, F. Castro and A. Juan. The RODRIGO database (submitted). *Proceedings of LREC 2010.* Valletta (Malta). May 2010.

GIDOC: Gimp-based Interactive transcription of old text DOCuments. A system prototype called GIDOC has been developed to provide user-friendly, integrated support for layout analysis, line detection and handwriting transcription. This work has led to four publications in international conferences:

- **ICIAP-2009:** L. Tarazón, D. Pérez, **N. Serrano**, V. Alabau, O. Ramos Terrades, A. Sanchis and A. Juan. Confidence Measures for Error Correction in Interactive Transcription of Handwritten Text. *Proceedings of the 15th ICIAP*. Vietri sul Mare (Italy). September 2009.

- **DRR-2010:** O. Ramos, **N. Serrano** and A. Juan. Interactive-predictive detection of handwritten text blocks (accepted). *Proceedings of the XVII DRR*. San Jose (USA). January 2010.
- **WEBIST-2010: N. Serrano**, L. Tarazón, D. Pérez, O. Ramos-Terrades and A. Juan. The GiDOC Prototype (submitted). *Proceeding of WEBIST 2010*. Valencia (Spain). April 2010.
- **CHI-2010: N. Serrano** and A. Juan. Demonstration of the GiDOC Prototype (submitted). *Media Showcase of CHI 2010*. Atlanta (USA). April 2010.

Adaptation and interaction in handwriting recognition. Using an interactive-predictive framework in old text transcription tasks, we have studied the effect of establishing a fixed degree of supervision. Moreover, we have proposed a simple yet effective method to find an optimal balance between recognition error and supervision effort. This work has led to two publications in international conferences:

- **ICMI-MLMI-2009: N. Serrano**, D. Perez, A. Sanchís and A. Juan. Adaptation from Partially Supervised Handwritten Text Transcriptions. *In Proceedings of the ICML-MLMI 2009*. Cambridge MA, USA. September 2009.
- **IUI-2010: N. Serrano**, A. Sanchís and A. Juan. Balancing Error and Supervision Effort in Interactive-Predictive Handwriting Recognition (Accepted) . *Proceedings of Intelligent User Interface 2010*. Hong-Kong, China. February 2010.

Kernel regression approach to machine translation. We presented a novel machine translation framework based on Kernel Regression techniques. Encouraging results have been obtained in a simple (but realistic) task. This work has led to a publication in an international conference:

- **IbPRIA-2009: N. Serrano**, J. Andrés-Ferrer and F. Casacuberta. On a Kernel Regression Approach to Machine Translation. *Proceedings of the 4th IbPRIA*. Póvoa de Varzim, Portugal. June 2009.

As said in the Introduction, it must be noted that the contributions described above are the result of a collaborative work involving other authors. The interested reader is referred to Table 1.1, for further information in the work attribution.

BIBLIOGRAPHY

[1] "iTransDoc: Interactive Transcription and Translation of Old Text Documents." `prhlt.iti.es/itransdoc.php.`, 2010.

[2] D. P. i Cardona, *Preparació de corpus i desenvolupament de prototips en reconeixement de text manuscrit.* PhD thesis, Dep. de Sistemes Informàtics i Computació, València, Spain, Dec 2009. Advisor(s): A. Juan and M. Pastor.

[3] L. T. Alcocer, *Confidence Measures in Interactive Handwritten Text Transcription.* PhD thesis, Dep. de Sistemas Informàticos i Computación, Valencia, Spain, Dec 2009. Advisor(s): A. Sanchís and A. Juan.

[4] U. V. Marti and H. Bunke, "The IAM-database: an English sentence database for off-line handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[5] T. Su, T. Zhang, and D. Guan, "Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text," *International Journal of Document Analysis and Recognition*, vol. 10, pp. 27–38, 2007.

[6] A. H. Toselli, V.Romero, L. Rodríguez, and E. Vidal, "Computer Assisted Transcription of Handwritten Text," in *Proccedings of 9th International Coference on Document Analysis and Recognition (ICDAR 2007)*, pp. 705–708, 2007.

[7] "The RODRIGO database: digitized data." `bvpb.mcu.es`, 2006.

[8] "The RODRIGO database: annotated data.." `prhlt.iti.es/rodrigo.php.`, 2010.

[9] A. Millares and J. M. Ruiz, *Tratado de paleografía española*, vol. 1. Espasa-Calpe, 3rd ed., 1983.

[10] D. Pérez, L. Tarazón, N. Serrano, F. Castro, O. Ramos, and A. Juan, "The GERMANA database," in *Proccedings of the 10th International Conference on Document Analysis and Recognition*, (Barcelona (Spain)), pp. 301–305, 2009.

[11] N. Serrano, L. Tarazón, D. Pérez, O. Ramos-Terrades, and A. Juan, "The GiDOC Prototype," in *Procceding of 6th International Conference on Web Information Systems and Technologies (WEBIST 2010)*, (Valencia (Spain)), 2010. (submitted).

[12] N. Serrano, F. Castro, and A. Juan, "The RODRIGO database," in *Proccedings of 7th Language Resources and Evaluation Conference (LREC 2010)*, (Valletta (Malta)), 2010. (submitted).

[13] `prhlt.iti.es/gidoc.php.`, 2009.

[14] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal on Document Analysis and Recognition*, vol. 9, pp. 123–138, 2007.

[15] O. Ramos, N. Serrano, and A. Juan, "Interactive-predictive detection of handwritten text blocks," in *Proccedings of the 17th Document Recognition and Retrieval Conference (DRR 2010)*, (San Jose, CA (USA)), 2010.

[16] A. H. Toselli, A. Juan, *et al.*, "Integrated handwriting recognition and interpretation using finite-state models," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 4, pp. 519–539, 2004.

[17] R. Bertolami and H. Bunke, "Hidden Markov model-based ensemble methods for offline handwritten text line recognition," *Pattern Recognition*, vol. 41, pp. 3452–3460, 2008.

[18] S. Young *et al.*, *The HTK Book*. Cambridge University Engineering Department, 1995.

[19] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proccedings of 7th the International Conference on Spoken Language Processing (ICSLP 2002)*, pp. 901–904, 2002.

[20] N. Serrano and A. Juan, "Demonstration of the GiDOC Prototype," in *Media Showncase of 28th ACM Conference on Human Factors in Computing Systems (CHI 2010)*, (Atlanta (USA)), 2010. (submitted).

[21] F. Wessel and H. Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 23–31, 2005.

[22] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Conf. measures for large vocabulary speech recognition.," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

[23] N. Serrano, D. Pérez, A. Sanchis, and A. Juan, "Adaptation from partially supervised handwritten text transcriptions," in *In Proccedings of the 11th International Conference on Multimodal Interfaces and the 6th Workshop on Machine Learning for Multimodal Interaction (ICML-MLMI 2009)*, (Cambridge, MA (USA)), pp. 289–292, 2009.

[24] N. Serrano, A. Sanchis, and A. Juan, "Balancing error and supervision effort in interactive-predictive handwriting recognition," in *Proccedings of 14th Inteligent User Interface (IUI 2010)*, (Hong Kong (China)), 2010. (Accepted).

[25] Z. Wang and J. Shawe-Taylor, "Kernel regression based machine translation," *NAACL HLT 2007 Companion Volume*, pp. 185–188, 2007.

[26] P. Koehn, F. Och, and D. Marcu, "Statistical phrase based translation," *In Proceedings of HLT/NACL*, 2003.

[27] C. Cortes, M. Mohri, and J. Weston, "A general regression technique for learning transductions," *Proccedings of the 22nd international conferences on Machine learning*, 2005.

[28] J. Gross and J. Yellen., "Handbook of graph theory," pp. 253–260, 2004.

[29] P. B. et al., "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistic*, vol. 19, no. 2, pp. 263–311, 1993.

[30] F. J. Och and H. Ney, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[31] P. Koehn, H. Hoang, A. Birch, and C. Callison-Burch., "Moses: Open source toolkit for statistical machine translation," *Proc. of ACL'07*, pp. 177–180, 2007.

[32] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady 10*, pp. 707–710, 1966.

[33] J. Goodman, "An empirical study of smoothing techniques for language modelling," *In Proccedings of ACL'96*, pp. 310–318, 1996.

[34] F. C. et al., "Some approaches to statistical and finite-state speech-to-speech translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[35] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[36] Papineni, K., Roukos, S., Ward, T., Zhu, and W. J, "Bleu: a method for automatic evaluation of machine translation," *Proccedings of ACL'02*, pp. 311–318, 2002.

[37] N. Serrano, J. Andrés-Ferrer, and F. Casacuberta, "On a kernel regression approach to machine translation," in *Proceedings of the 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2009)*, vol. 5524 of *LNCS*, (Póvoa de Varzim (Portugal)), pp. 394–401, Springer-Verlag, June 2009.

LIST OF FIGURES

LIST OF TABLES