

UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN



Multimodal Interactive Transcription of Handwritten Text Images

Thesis
presented by Verónica Romero Gómez
supervised by Prof. Enrique Vidal Ruiz and Dr. Alejandro Héctor Toselli Rossi

September 3, 2010

Multimodal Interactive Transcription of Handwritten Text Images

Verónica Romero Gómez

Thesis performed under the supervision of doctors
Enrique Vidal Ruiz and Alejandro Héctor Toselli Rossi
and presented at the Universidad Politécnica de Valencia
in partial fulfilment of the
requirements for the degree
Doctor en Informática

Valencia, September 3, 2010

Work supported by the Spanish Government (MICINN and “Plan E”) under the MITTRAL (TIN2009-14633-C03-01) research project and under the research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), and by the Universidad Politécnica de Valencia (FPI fellowship 2006-04).

Acknowledgements

Desde el momento que decidí dedicarme a la investigación y llevar a cabo un doctorado, muchas son las personas que de una manera u otra han influido en la realización del presente trabajo. Personas que me han ayudado y a las que me gustaría darles las gracias por todo el esfuerzo, el tiempo y la dedicación invertida.

En primer lugar me gustaría darles las gracias a mis directores de tesis Enrique Vidal y Alejandro Héctor Toselli, quienes con su esfuerzo, enseñanzas y consejos han conseguido que esta tesis sea una realidad. Gracias por todas las horas dedicadas y por su predisposición permanente e incondicional a aclarar mis dudas y prestarme su ayuda. Esta tesis es tan suya como mía. También quiero darle las gracias a Moisés, quien tanto me ayudo cuando empecé y me enseñó todos los entresijos del reconocimiento de texto manuscrito y quien siempre ha estado dispuesto a echarme una mano.

Un agradecimiento especial merecen todos mis compañeros del PRHLT. Gente en el ITI y en el DSIC con la que he compartido muy buenos momentos y que me han proporcionado un entorno estimulante para llevar a cabo este trabajo. Empezaré por la gente del DSIC, y más concretamente por los habitantes del laboratorio 105, lugar donde empecé mi labor investigadora y donde encontré gente excepcional que me ha ayudado a crecer como profesional y como persona. Gracias a Miriam Luján por su amistad y apoyo; a Vicent Tamarit por dejarme utilizar su cuenta en hyades, a Guillem Gascó por su compañía mientras estuvimos en Aachen; a Germán Sanchis por las interesantes conversaciones mantenidas; a Marta y Pascual por su amabilidad y muy especialmente a Vicent Alabau, quien siempre ha estado dispuesto a ayudarme, sin quejas, sino más bien todo lo contrario, con ese humor que le caracteriza.

También en el ITI me he encontrado con gente estupenda con la que he compartido buenos momentos y que de alguna manera a contribuido a esta tesis. Me gustaría nombrar a Jesús González, Antonio Lagarda, Daniel Ortiz, Jose Ramón Navarro, Nicolás Serrano, Lionel Tarazón, Oriol Terrades y Isaías Sánchez. También quiero darle las gracias a Luis Leiva, por nuestra colaboración en el desarrollo de la interfaz de CATTI, que hoy forma parte de esta tesis; a Jesús Andrés, por prestarme la plantilla de esta tesis que tantos dolores de cabeza me ha ahorrado; y a Jorge Civera, por sus consejos y su ayuda, y por prestarme el software que ha servido de base de esta tesis.

En el ámbito personal quiero empezar dándoles las gracias a mis padres, a los que nunca podré agradecer suficiente todo lo que han hecho por mí. Ellos han estado siempre a mi lado compartiendo mis alegrías y mis penas, dándome su apoyo, su confianza y su amor incondicional. Que estas palabras sirvan de agradecimiento a toda una vida de dedicación. También quiero darle las gracias a mi hermano por ser mucho más que eso, por ser mi amigo, mi confidente y mi modelo a seguir, por estar siempre dispuesto a ayudarme, preocuparse por mí y darme su apoyo; a mi cuñada a la que conozco ya tanto tiempo que es para mí como una hermana; a mi abuela que siempre ha estado a mi lado, y a Elena, que acaba de llegar y ya ha llenado mi casa de alegría.

No puedo olvidarme de mis amigos, aquellos que conozco prácticamente desde que nací, y aquellos que he ido haciendo a lo largo del camino. Personas con las que he crecido, con las que he compartido momentos muy importantes de mi vida y que han estado conmigo en los buenos y en los malos momentos. Amigos que, aun sin acabar de entender muy bien a que me dedico, siempre han confiado en mí, con la seguridad de que podría conseguir todo lo que me propusiera.

Por último, quiero darle las gracias a César, por compartir su vida conmigo y aceptarme tal y como soy. Por aguantar mis nervios, mis manías y mi estrés, sobre todo en esta última fase de la tesis, y saber siempre hacerme sonreír. Por cuidarme, quererme y hacerme feliz.

A todos mi más sincera gratitud.

Verónica Romero Gómez
Valencia, 27 de julio de 2010

This thesis presents an interactive multimodal approach for efficient transcription of handwritten text images. This approach, rather than full automation, aims at assisting the expert in the proper recognition transcription process.

Until now, the handwritten text recognition systems (HTR) are far from being perfect and heavy human intervention is often required to check and correct the results of such systems. HTR systems have indeed proven useful for restricted applications involving form-constrained handwriting and/or fairly limited vocabulary (such as postal addresses or bank check legal amounts), achieving in this kind of tasks relatively high recognition accuracy. However, in the case of unconstrained handwritten documents (such as old manuscripts and/or unconstrained, spontaneous text), current HTR technology typically only achieves results which are far from being directly acceptable in practice.

The interactive scenario studied in this thesis allows for a more effective approach. Here, the automatic HTR system and the human transcriber cooperate to generate the final transcription of the text images. In this scenario, the system uses the text image and a previously validated part (prefix) of its transcription to propose a suitable continuation. Then the user finds and corrects the next system error, thereby providing a longer prefix which the system uses to suggest a new, hopefully better continuation. The technology used in this work is based on Hidden Markov Models (HMMs) and n -gram language models, used in the same way as they are used in the current automatic speech recognition (ASR) systems. To take into account the feedback introduced by the user some modifications in the conventional n -gram language models have been studied. To implement the decoding process in one step, as in conventional HTR systems, two main approaches are presented. The first of them consist in building a special language model and the second one, on more sophisticated word-graph techniques. The last approach integrates efficient error-correcting algorithms in order to guarantee low response time and preserve adequate transcription accuracy. The system was tested on three corpora, two of them contain handwritten text in modern Spanish and English, whereas, the third corpus consists of cursive handwritten page images in old Spanish. The results on the three cursive handwritten tasks suggest that, using the interactive approach, considerable amounts of user effort can be saved with respect to both pure manual work and

non-interactive HTR systems.

In the interactive system presented here the user is repeatedly interacting with the system. Hence, the quality and ergonomics of the interactive process is crucial for the success of the system. In this thesis, different ways to interact with the system and different levels (whole word and keystroke) have been studied. Moreover, more ergonomic multimodal interfaces have been used in order to obtain an easier and more comfortable human-machine interaction. Among many possible feedback modalities, we focus here on touchscreen communication, which is perhaps the most natural modality to provide the required feedback. The on-line feedback HTR subsystem used is based on HMMs in the same way that the main off-line HTR system. To train the on-line HTR feedback subsystem and test the multimodal approach, an on-line handwriting corpus has been used. The required word instances that would have to be handwritten by the user in the multimodal interaction process were generated by concatenating random character instances from three categories: digits, lowercase letters and symbols. The obtained results show that, in spite of losing the deterministic accuracy of the traditional keyboard and mouse, the more ergonomic multimodal approach can save significant amounts of human effort.

En esta tesis se presenta un nuevo marco interactivo y multimodal para la transcripción de documentos manuscritos. Esta aproximación, lejos de proporcionar la transcripción completa pretende asistir al experto en la dura tarea de transcribir.

Hasta la fecha, los sistemas de reconocimiento de texto manuscrito disponibles no proporcionan transcripciones aceptables por los usuarios y, generalmente, se requiere la intervención del humano para corregir las transcripciones obtenidas. Estos sistemas han demostrado ser realmente útiles en aplicaciones restringidas y con vocabularios limitados (como es el caso del reconocimiento de direcciones postales o de cantidades numéricas en cheques bancarios), consiguiendo en este tipo de tareas resultados aceptables. Sin embargo, cuando se trabaja con documentos manuscritos sin ningún tipo de restricción (como documentos manuscritos antiguos o texto espontáneo), la tecnología actual solo consigue resultados inaceptables.

El escenario interactivo estudiado en esta tesis permite una solución más efectiva. En este escenario, el sistema de reconocimiento y el usuario cooperan para generar la transcripción final de la imagen de texto. El sistema utiliza la imagen de texto y una parte de la transcripción previamente validada (prefijo) para proponer una posible continuación. Después, el usuario encuentra y corrige el siguiente error producido por el sistema, generando así un nuevo prefijo más largo. Este nuevo prefijo, es utilizado por el sistema para sugerir una nueva hipótesis. La tecnología utilizada se basa en modelos ocultos de Markov y n -gramas. Estos modelos son utilizados aquí de la misma manera que en el reconocimiento automático del habla. Algunas modificaciones en la definición convencional de los n -gramas han sido necesarias para tener en cuenta la retroalimentación del usuario en este sistema. Para implementar el proceso de decodificación en un solo paso, tal y como se hace en los sistemas convencionales de reconocimiento de texto, dos aproximaciones han sido estudiadas. La primera de ellas consiste en la construcción de un modelo de lenguaje especial, y la segunda se basa en la utilización de grafos de palabras. En esta última aproximación, se integran técnicas eficientes de corrección de errores, con el fin de garantizar el bajo tiempo de respuesta y un mínimo de precisión en las transcripciones. La calidad del sistema ha sido medida automáticamente con tres corpus. Dos de ellos contienen texto manuscrito en español e inglés moderno, mientras que, el tercer corpus consiste en un texto manuscrito antiguo del siglo XIX. Los resultados obtenidos con

los tres corpus sugieren que utilizar el escenario interactivo propuesto puede reducir considerablemente el esfuerzo realizado por el usuario si lo comparamos con el sistema convencional de reconocimiento de texto.

En este nuevo escenario interactivo, el usuario esta repetidamente interactuando con el sistema, por lo tanto, la calidad y ergonomía en el proceso interactivo es crucial para su éxito. En esta tesis, se han estudiado diferentes formas de interactuar con el sistema y diferentes niveles de interacción (palabras completas o caracteres). Además, se han utilizado interfaces multimodales más ergonómicas, con la intención de obtener sistemas más confortables y de fácil uso para el usuario. Entre todas las posibles modalidades de retroalimentación, nos centramos en la comunicación mediante una pantalla táctil, que es, quizás, la forma más natural de proporcionarle al sistema la retroalimentación. El subsistema de reconocimiento de texto manuscrito on-line utilizado para la decodificación de la retroalimentación también esta basado en modelos ocultos de Markov. Para entrenar los modelos del subsistema de retroalimentación, y testear la versión multimodal del escenario interactivo, se ha utilizado un corpus manuscrito on-line. Las palabras que debe introducir el usuario en el proceso multimodal han sido generadas concatenando muestras aleatorias de caracteres de tres categorías diferentes: dígitos, letras minúsculas y símbolos. Los resultados obtenidos muestran que, a pesar de la pérdida del determinismo que proporcionan el teclado y el ratón, la aproximación multimodal puede ahorrar una cantidad significativa de esfuerzo humano.

En aquesta tesi es presenta un nou marc interactiu i multimodal per a la transcripció de documents manuscrits. Aquesta aproximació, lluny de proporcionar la transcripció completa pretén assistir a l'expert en la tasca de transcriure.

Fins ara, els sistemes de reconeixement de text manuscrit que hi ha disponibles no proporcionen transcripcions acceptables pels usuaris i, generalment, es necessària la intervenció de l'humà per corregir les transcripcions obtingudes. Aquests sistemes han demostrat ser realment útils en aplicacions restringides amb vocabularis limitats (com es el cas del reconeixement de direccions postals o de quantitats numèriques en xecs bancaris), aconseguint en aquest tipus de tasques resultats acceptables. No obstant això, si es treballa amb documents manuscrits sense cap tipus de restricció (com documents manuscrits antics o text espontani), la tecnologia actual sols aconsegueix resultats inacceptables.

L'escenari interactiu estudiat en aquesta tesi permet una solució més efectiva. En aquest escenari, el sistema de reconeixement i l'usuari cooperen per generar la transcripció final de la imatge de text. El sistema utilitza la imatge de text i una part de la transcripció prèviament validada (prefix) per proposar una possible continuació. Després, l'usuari troba i corregeix el següent error produït pel sistema, generant així un nou prefix més llarg. Aquest nou prefix, és utilitzat pel sistema per suggerir una nova hipòtesi. La tecnologia utilitzada es basa en models ocults de Markov i n -grames. Aquests models són utilitzats ací de la mateixa manera que en el reconeixement automàtic de la parla. Algunes modificacions en la definició convencional dels n -grames han sigut necessàries per tindre en compte la retroalimentació de l'usuari en el sistema. Per altre costat, per implementar el procés de descodificació en un únic pas, tal i com es fa en els sistemes convencionals de reconeixement de text, dos aproximacions han sigut estudiades. La primera d'elles consisteix en la construcció d'un model de llenguatge especial, i la segona es basa en la utilització de grafs de paraules. En aquesta última aproximació, s'integren eficients tècniques de correcció d'errors, amb l'objectiu de garantir el baix temps de resposta i un mínim de precisió en les transcripcions. La qualitat del sistema ha sigut mesurada automàticament amb tres corpus. Dos d'ells contenen text manuscrit en espanyol i anglès modern, mentre que l'altre, consisteix en un text manuscrit antic del segle XIX. Els resultats obtinguts amb els tres corpus mostren que utilitzar l'escenari

interactiu proposat pot reduir considerablement l'esforç realitzat pel usuari si el comparem amb el sistema convencional de reconeixement de text.

L'usuari esta repetidament interactuant amb el sistema proposat, per això, la qualitat i l'ergonomia en el proces interactiu és crucial per a l'èxit del sistema. En aquesta tesi, s'han estudiat diferents formes d'interactuar amb el sistema i diferents nivells d'interacció (paraules completes o caràcters). A més, s'han utilitzat interfícies multimodals més ergonòmiques, amb la intenció d'obtindre sistemes més confortables i de fàcil ús per al usuari. Entre totes les possibles modalitats de retroalimentació, ens centrem en la comunicació utilitzant una pantalla tàctil, que és, possiblement, la forma més natural de proporcionar, al sistema, la retroalimentació. El subsistema de reconeixement de text manuscrit on-line utilitzat per a la descodificació de la retroalimentació introduïda pel usuari, es basa en l'ús de models ocultes de Markov, de la mateixa manera que en el sistema principal. Per entrenar els models del subsistema de retroalimentació, i testar la versió multimodal de l'escenari interactiu, s'ha utilitzat un corpus manuscrit on-line. Les paraules que han de ser introduïdes per l'usuari en el procés multimodal han sigut generades concatenant mostres aleatòries de caràcters de tres categories diferents: dígit, lletres minúscules i símbols. Els resultats obtinguts mostren que, tot i la pèrdua del determinisme que proporcionen el teclat i el ratolí, l'aproximació multimodal pot estalviar una quantitat significativa d'esforç humà.

The Handwritten Text Recognition (HTR) is not a new field of pattern recognition. In fact, HTR had its beginnings in the late 1960s with the recognition of zip codes on letters or account and amount information on checks. However, the computer capacity of those days was not enough for large scale HTR. It is now, when this task is becoming an important research topic. This is mainly due to two reasons, the increasing in the computer capacity and, specially because the huge amount of text that is only available in handwriting, for example, almost all historical documents. Many of these documents images need to be transcribed in order to provide historians, other researchers and the general public new ways of indexing, consulting and querying them.

These transcriptions are usually carried out by paleography experts, who are specialized in reading ancient scripts, characterized, among other things, by different handwritten/printed styles from diverse places and time periods. How long experts takes to make a transcription of one of these documents depends on their skills and experience, as well as on the type of text and the quality of the documents to be transcribed. In some cases it may take several hours per page.

On the other hand, up-to-date handwritten text recognition system (HTR) cannot substitute the experts in this task, because there are no efficient solutions to perform them automatically with a good accuracy. HTR systems have indeed proven useful for restricted applications involving form-constrained handwriting and/or fairly limited vocabulary (such as postal addresses or bank check legal amounts), achieving in this kind of tasks relatively high recognition accuracy. However, in the case of unconstrained handwritten documents (such as old manuscripts and/or unconstrained, spontaneous text), current HTR technology typically only achieves results which are far from being directly acceptable in practice. The difficulties to segment text lines, the variability of the handwriting, the complexity of the styles and the open vocabulary explain most of the issues encountered by these recognition systems.

Therefore, once a full recognition process of the document has finished, heavy human-expert revision is required to really produce a transcription of standard quality. The human transcriber is, therefore, responsible for verifying and correcting the mistakes made by the

system. Given the high error rates involved, such a “*post-editing*” solution is quite inefficient and uncomfortable for the human corrector, who sometimes prefers work without any kind of HTR system.

An interactive scenario, where the automatic HTR system and the human transcriber cooperate to generate the final transcription of the text images, can allow for a more effective approach. The rationale behind this approach is to combine the accuracy provided by the transcriber expert with the efficiency of the HTR system. This scenario follows the idea of developing machines with the aim of assisting human beings in their works, instead of fully automatic devices. This idea has been previously applied to machine translation and speech transcription, where experiments and real field tests have shown that, by capitalizing on the human feedback, this kind of systems can save significant amounts of overall human effort.

The main objective of this thesis is to put forward the theoretical framework, as well as to develop and to assess a *multimodal computer assisted transcription system for handwritten text images*. More precisely, the scientific objectives of this thesis can be divided into four groups as follows:

1. **Computer assisted transcription of handwritten text images.** One of the main objectives of this thesis is to extend the interactive predictive scenario, previously applied to machine translation and speech transcription, to the handwritten text recognition problem.
2. **Ergonomics.** The user is repeatedly interacting with the computer assisted transcription of handwritten text images system. So, one of the objectives of this thesis is to make the interaction process friendly and ergonomic to the user.
3. **Response time.** The response time is a critical feature of the system. The system must be able to interact with the human expert in a time efficient way, otherwise the user will prefer to transcribe the document without any help. In this work we will study different implementations trying to obtain a good response time without losing too much accuracy.
4. **Multimodal computer assisted transcription of handwritten text images.** To introduce more ergonomic multimodal interfaces should result in an easier and more comfortable human-machine interaction. In this thesis we intend to develop and to assess a multimodal computer assisted transcription system which focus on touchscreen communication.

The above objectives are sequentially studied in 6 chapters that cover most of the work developed in this thesis. In Chapter 1, we introduce the problem that we are trying to solve and then, we overview the state-of-the-art of the OCR and HTR and the formal Pattern Recognition (PR) framework for multimodal interaction. Later, the bases, formulations and algorithms related with the HMMs, n -gram language models and word-graphs are presented.

In Chapter 2 the four data sets used on the experiments are introduced. We use three off-line handwritten data sets and one on-line handwritten corpus. The first off-line data set is a corpus based on a realistic application: transcription of handwritten answers from survey forms in modern Spanish. The second one consist of handwritten full sentences in modern English. And the last one, was compiled from a legacy handwriting document written on the

XIX century. On the other hand, the on-line corpus is an English dataset divided into several categories: letters, digits, symbols, isolated words and full sentences.

A detailed description of the *off*- and *on*-line HTR systems is given in Chapter 3. Both the *off* and the *on*-line systems follow the classical architecture composed of three modules: preprocessing, feature extraction and recognition. All these modules will be explored and assessed with the corpora described on Chapter 2.

In Chapter 4, a computer assisted transcription of handwritten text images system is introduced and assessed on the three off-line handwritten corpora presented in Chapter 2. In order to improve the quality and ergonomy of the interactive process different ways to interact with the system using the keyboard and the mouse are studied. In addition, the character level interaction is developed. To improve the response time, two different approaches to implement the decoding process are also derived and assessed on the same off-line handwritten tasks.

The multimodal version of the CATTI system is presented and automatically evaluated in Chapter 5. A touchscreen communication is studied in order to obtain an easier and more comfortable human-machine interaction. The on-line feedback HTR subsystem is based on HMMs and n -gram language models. The different ways to join the off-line and the on-line information are assessed using both, the off-line and the on-line dataset presented on Chapter 2.

In Chapter 6 implementation details of the approaches studied in previous sections are presented. In particular, a web-based demonstrator for interactive transcription of handwritten text images is introduced.

In Chapter 7, a summary of the work and contributions presented in this thesis are discussed, followed by an outlook.

Finally, a list of mathematical symbols and acronyms used throughout this thesis is presented in Appendix A. Additional experiments of the off-line HTR system are presented in the Appendix B.

CONTENTS

Acknowledgements	v
Abstract	vii
Resumen	ix
Resum	xi
Preface	xiii
Contents	xvii
1 Preliminaries	1
1.1 Introduction	1
1.2 State of the art	4
1.2.1 Optical Character Recognition	6
1.2.2 Handwritten Text Recognition	6
1.3 Interactive Pattern Recognition	7
1.4 Theoretical Background	8
1.4.1 Hidden Markov Models	9
1.4.2 Language models: <i>N</i> -grams	14
1.4.3 Word-graphs	17
1.5 Scientific objectives	21
Bibliography	23
2 Corpora	29
2.1 Introduction	29
2.2 Cristo Salvador	29
2.3 ODEC	32

2.4	IAM database	34
2.5	UNIPEN	37
	Bibliography	39
3	Handwritten Text Recognition	41
3.1	Introduction	41
3.2	Off-line Handwritten Text Recognition	43
3.2.1	Preprocessing	43
	Background removal and noise reduction	43
	Skew correction	43
	Line extraction	44
	Slope correction	45
	Slant correction	46
	Size normalization	46
3.2.2	Feature extraction	47
3.2.3	Recognition	48
	Probabilistic Framework	48
	Character, Word and Language Modelling	49
3.2.4	Experimental Framework	52
	Corpora	52
	Assessment Measures	52
	Parameters	53
3.2.5	Results	54
3.2.6	Summary and Conclusions	59
3.3	On-line Handwritten Text Recognition	59
3.3.1	Preprocessing	60
3.3.2	Feature Extraction	60
3.3.3	Recognition	61
3.3.4	Experimental Framework	61
	Corpora	61
	Assessment Measures	62
	Parameters	62
3.3.5	Results	62
3.3.6	Summary and Conclusions	62
	Bibliography	63
4	Computer Assisted Transcription of Handwritten Text Images	67
4.1	Introduction	67
4.2	Formal Framework	69
4.3	Adapting the Language Model	70
4.4	Searching	71
4.4.1	Viterbi-based approach	71
4.4.2	Word-graph based approach	72
	Error-correction parsing	74
4.5	Increasing interaction ergonomoy	76

4.5.1	Language Model and Search	78
4.6	CATTI at the character level	78
4.6.1	Language Model and Search	80
4.7	Experimental framework	81
4.7.1	Assessment Measures	82
4.7.2	Parameters	83
4.8	Results	83
4.9	Conclusions and future work	91
	Bibliography	93
5	Multimodal Computer Assisted Transcription of Handwritten Text Images	95
5.1	Introduction	95
5.2	Formal Framework	97
5.3	Adapting the Language Model	99
5.4	Searching	100
5.5	Experimental Framework	101
5.5.1	Corpora	101
5.5.2	Assessment Measures	102
5.6	Results	102
5.7	Conclusions	105
	Bibliography	107
6	A Web-based Demonstrator to Interactive Multimodal Transcription	109
6.1	Introduction	109
6.2	User Interaction Protocol	110
6.3	System description	110
6.3.1	Application Programming Interface	111
6.3.2	MM-CATTI server	112
6.3.3	Web Interface	112
6.3.4	Electronic Pen or Touchscreen Interaction	114
6.3.5	Keyboard Interaction	115
6.4	Results and Conclusions	118
	Bibliography	121
7	Conclusions and Future Work	123
7.1	Conclusions	123
7.2	Publications related with this work	125
7.3	Future work	128
	Bibliography	131
A	Symbols and Acronyms	133
A.1	Symbols	133
A.2	Acronyms	136

B Additional Experiments on Off-line Handwritten Text Recognition	137
B.1 Cristo-Salvador corpora	137
B.2 ODEC corpora	140
B.3 IAMDB corpora	141
B.4 Summary	143
Bibliography	145
List of Figures	147
List of Tables	153

CHAPTER *1*

Preliminaries

1.1 Introduction

Lately, the paradigm for Pattern Recognition (PR) systems design has been shifting from the concept of full-automaton, i.e. systems where no human intervention is assumed, to systems where the decision process is affected by human feedback. One remarkable PR example where this feedback can be successfully used is handwritten document transcription. This task is becoming an important research topic, specially because of the increasing number of on-line digital libraries publishing large quantities of digitized legacy documents. The vast majority of these documents, hundreds of terabytes worth of digital image data, remain waiting to be transcribed into a textual electronic format that would provide historians and other researchers new ways of indexing, consulting and querying these documents.

These transcriptions are usually carried out by experts in paleography, who are specialized in reading ancient scripts, characterized, among other things, by different calligraphy/print styles from diverse places and time periods. How long experts take to carry out a transcription of one of these documents depends on their skills and experience, as well as on the type of text and the quality of the documents to be transcribed. For example, to transcribe many of the pages of one of the documents used in this thesis, they would typically spend more than half an hour per page.

State-of-the-art cursive handwritten text recognition systems (HTR) can by no means substitute the experts in this task. HTR systems have indeed proven useful for restricted applications involving form-constrained handwriting and/or fairly limited vocabulary (such as postal addresses or bank check legal amounts), achieving in this kind of tasks relatively high recognition accuracy [SK97, DIMP02]. However, in the case of unconstrained handwritten

documents (such as old manuscripts and/or unconstrained, spontaneous text), current HTR technology typically only achieves results which are far from being directly acceptable in practice.

Therefore, once the full recognition process of one of these documents has finished, heavy human expert revision is required to really produce a transcription of standard quality. The human transcriber is, therefore, responsible for verifying and correcting the mistakes made by the system. In this context, the HTR process is performed off-line. First, the HTR system returns a full transcription to all the text lines in the whole document. Next, the human transcriber reads this sequentially (while looking at their correspondence in the original page images) and corrects the possible mistakes made by the system. Given the high error rates involved, such a *post-editing* solution is quite inefficient and uncomfortable for the human corrector, who often prefers to carry out the transcription without using any kind of HTR system.

An *interactive* scenario would allow for a more effective approach. This thesis aims to develop innovative technologies to implement *computer assisted* solutions. These technologies are based on a recently introduced framework called “interactive predictive” (IP) processing (IPP) [VRCGV07]. In this framework the automatic HTR system and the human transcriber cooperate to generate the final transcription of the text images. The rationale behind this approximation is to combine the accuracy provided by the transcription expert with the efficiency of the HTR system. We call this approach “Computer Assisted Transcription of Text Images” (CATTI). It follows similar ideas as those previously applied to computer assisted translation [CVC+04, BBC+09] and speech transcription [RCV07, VRCGV07], where experiments and real field tests have shown that, by capitalizing on the human feedback, this kind of systems can save significant amounts of overall human effort. In this approach, at each interaction step, the system proposes its best output for the given input data (e.g., its best transcription for the given text image). If the user finds it correct, then it is accepted and the process goes on with successive input data. Otherwise, the user introduces some information that the system takes into account in order to improve the proposed transcription.

This scenario is fundamentally different from the (generally unsatisfactory) non-interactive post-editing solution in at least two relevant aspects. First, the fact that the user will be involved on the transcription process provides a much more friendly environment, letting the user be in command of the system, rather than the other way around. And second, the effort needed by the user to obtain the perfect transcription of the input handwritten text image can be significantly smaller. It is thanks to the direct payoff obtained by taking advantage of the user feedback information to immediately improve system results. That is, when the user amends some erroneous element found in the system output, the system reacts with a revised output where not only this error is fixed, but other subsequent or related errors can be corrected.

Another important aspect of this work is *multimodal processing*. As will be discussed later, human feedback signals in interactive systems rarely belong to the same domain as the one the main data stream comes from, thereby entailing some sort of *multimodality*. Of course, this is the case in CATTI, where the main data are text images and feedback would consist of keystrokes and/or pointer positioning actions. Nevertheless, at the expense of losing the deterministic accuracy of the traditional keyboard and mouse, more ergonomic multimodal interfaces are possible. It is worth noting, however, that the potential increase in user-



Figure 1.1: Left: illustration of CATTI multimodal user-interaction using a touch-screen. Right: page fragment showing a line image being processed, with a partially corrected system suggestion (in grey and black roman font) and the (previous) corrections made by the user through pen strokes and handwriting input marked in red.

friendliness comes at the cost of acknowledging new possible errors coming from the decoding of the feedback signals. Therefore, solving the *multimodal interaction problem* amounts to achieving a *modality synergy* where both main and feedback data streams help each-other to optimize overall accuracy. These ideas have recently been explored in the context of computer assisted translation, using speech signals for feedback [VCR+07, VRCGV07].

Among many possible feedback modalities for CATTI, we focus here on touch-screen operation, which is perhaps the most natural modality to provide the required feedback in CATTI systems. Figure 1.1 (left) shows a user interacting with a CATTI system by means of a touch-screen. Both the original text image and the successive *off-line* HTR system’s transcription hypotheses can be easily aligned and jointly displayed on the touchscreen, as shown in Figure 1.1 (right). This way, the user corrective feedback can be quite naturally provided by means of pen strokes, exactly registered over the text produced by the system, which are fed to an *on-line* HTR subsystem. We will use the shorthand “MM-CATTI” for this kind of *multimodal* CATTI processing. Touchscreen devices are very popular human-computer interfaces for editing tasks. For instance, in [SMW01] and [LS06], respectively, they are considered for (non-interactive) post-editing and for interactively correcting the output of a speech recognizer.

A uniform technology, based on Hidden Markov Models (HMMs) is used in this work both for the main *off-line* HTR system and for the *on-line* feedback HTR subsystem. HMMs are used in the same way as they are used in the current automatic speech recognition (ASR) systems [Rab89]. The most important differences lay in the type of input feature vectors sequences; while they represent acoustic data in the case of ASR, line-image features and point coordinates of handwritten pen strokes constitute the input sequences for off- and on-line HTR, respectively.

In this thesis we study the CATTI framework and its multimodal version, MM-CATTI. We also carry out a comprehensive series of experiments to assess the capabilities of CATTI and MM-CATTI.

Clearly, in these interactive scenarios, assessing system performance should ultimately require human work and judgement. However, this human judgement is prohibitively expen-

sive, because, in order to assess the validity of our assumptions and estimations, it will not be enough with just one transcriber expert, but an entire panel of transcribers will be required. Each one of these transcribers ought to carry out several trials, where each trial round should include at least one dry-run session, during which the participant is asked to transcribe some document using the same interface but without the benefit of the system's predictions. And it must be taken into account that, using the same document to transcribe in both sessions would be unfair, because the user would tend to familiarize herself with the text during the dry-run session, thereby apparently achieving much higher productivity in the second session. However, if different texts are used the results will not be comparable. Consequently, the best solution is to work with the same document but allowing a lot of time to pass between experiments. In addition, for these experiments to lead reliable results, appropriate interfaces for professional use must be implemented. This kind of experiments were carried out in the TT2 project [SdIfIV+] in which Interactive-Predictive approaches were developed and tested for high-quality Computer-Assisted Translation. While the corpus-based, objective *estimated* effort reduction measures did provide useful research feedback information [BBC+09], only rather vague, mostly qualitative conclusions could be finally derived from the (otherwise extremely expensive) experiments with real users [CCC+09].

In conclusion, to carry out a reliable test with real users is much too expensive for the current stage of the developments and, on the other hand, it could by itself become the focus of a different PhD work.

Fortunately, the very convenient and successful PR assessment paradigm based on labelled corpora is still applicable here to obtain adequate *estimates* of human effort required to achieve the goals of the considered tasks. To this end, corpora corresponding to three handwritten text transcription tasks are considered, including the well-known and publicly available IAMDB corpus [MB99].

In this chapter, first we overview the state-of-the-art of OCR and HTR systems in Section 1.2. Then, we focus on Interactive Pattern Recognition on Section 1.3. Next, the bases, formulations and algorithms related with the HMMs, the n -grams and the word-graphs are enunciated on Section 1.4. Finally, we summarize the scientific objectives of this thesis in Section 1.5.

1.2 State of the art

Many documents used every day are handwritten documents, as for example, postal addresses, bank cheques, medical prescriptions, a big quantity of historical documents, an important part of the information gathered by forms, etc. In many cases it would be interesting to have these documents in digital form rather than paper based, in order to provide new ways to indexing, consulting and working with these documents.

Handwriting text recognition (HTR) can be defined as the ability of a computer to transform handwritten input represented in its spatial form of graphical marks into equivalent symbolic representation as ASCII text. Usually, this handwritten input comes from sources such as paper documents, photographs or electronic pens and touch-screens.

HTR is a relatively new field of computer vision. The optical character recognition (OCR) had its beginnings in the year 1951 with the optical reader invented by David H. Shep-

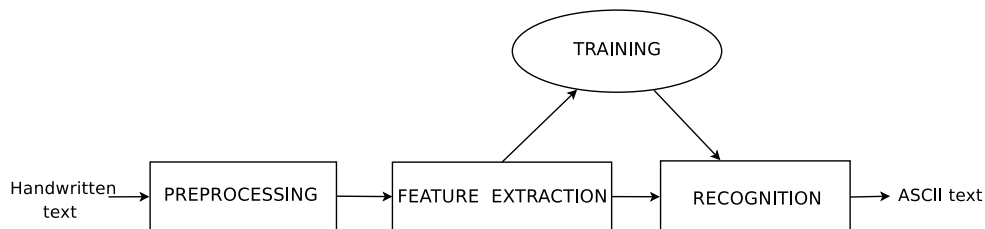


Figure 1.2: Diagram representing the different modules of an handwritten text recognition system.

ard. This reader, known as GISMO, was able to read typewritten text, Morse and musical notes [She53]. HTR first appeared in the late 1960s with restricted applications with very limited vocabulary, such as the recognition of postal addresses envelopes or the recognition of bank cheques. The computer capacity and the available technology of those days was not enough for unconstrained handwritten documents (such as old manuscripts and/or unconstrained, spontaneous text). Even the recognition of printed text was only adequate on simplified fonts, for example the font used on credit cards ever since. However, the increase in computers capacity of these days, the development of adequate technologies and the necessity of processing automatically a big quantity of handwritten documents have converted handwritten text recognition in a focus of attention both from industry as well as the research community, leading to a significant improvement of the accuracy of these systems.

According to [BRST95, BB08], nowadays, OCR systems are capable of accurately recognizing typewritings in several fonts. However, high recognition accuracy is still difficult to achieve in continuous handwritten text. The reason for this has to be seen in the high variability such as individual writing styles, the number of different word classes and the word segmentation problem, which comes with handwriting.

Essentially, the HTR systems follow the classical pattern recognition (PR) architecture composed of three modules (see Figure 1.2):

- **Acquisition and preprocessing module:** this module is in charge to acquire data and preprocessing it. Here the noise of the data is filter out, the handwritten strokes from degraded images are recover and the variability of the text styles is reduced.
- **The feature extraction module:** in this module a feature vector sequence is obtained as the representation of the handwritten text.
- **The recognition module:** here the most likely word sequence for the sequence of feature vectors is obtained.

All HTR systems involve two different phases: training and recognition. Usually, training is carried out in the laboratory, where the different preprocessing operations, the different features and the classification methods are studied in order to choose those that obtain better results. If the labelled of the samples used for training are known we speak of *supervised learning*. However, if only unlabelled examples are given the training is called *unsupervised*

learning. Unlike training, recognition is the operative phase of the HTR system, and it is in charge to recognize all the unknown samples that arrive to the system.

Current research in HTR mainly follows two different basic approaches which will be discussed in the next subsections. The first approach recognizes isolated characters or digits. Therefore, when dealing with non-isolated characters, for example in words or text, the data has to be segmented beforehand. The origin of this method has to be seen in the beginning of HTR when the input images contained only one single symbol. The second approach uses a continuous recognition, meaning that beginning or ending of each character are not needed before hand and they are hypothesized as a byproduct of the proper recognition process.

1.2.1 Optical Character Recognition

Optical character recognition, usually abbreviated to OCR, is the mechanical translation of images of handwritten, typewritten or printed text (letters, numbers or symbols) into machine-editable text. An historic review of OCR can be found on [MSY92, IOO91]. OCR is mainly divided in two areas: printed character or word recognition, and handwritten character or word recognition. A big improvement on the recognition of printed text has been obtained until now, mainly due to the fact that is easy to segment the text into characters. This improvement has made possible the development of OCR systems with a very good accuracy, around 99% at the character level. However, given the kind of text image documents involved in this thesis, OCR products, or even the most advanced OCR research prototypes [Rat03], are very far from offering useful solutions to the transcription problem. They are simply not usable since, in the vast majority of the handwritten text images of interest, characters can by no means be isolated automatically.

In OCR field the recognition is performed using one or several classifiers. A naive approach to the recognition of an image containing a single symbol is the nearest neighbour (NN) or k-NN approach. More sophisticated algorithms which achieve better results are artificial neural networks (ANNs) and support vector machines (SVMs) [Lee96, Sri93]. Moreover, some work has been carried out using tangent vectors and local representations [KPNV02] or Bernoulli mixture models [JV01, RGJ07, AGJ10] with good results. On [FLS92, Nag00] it is carried out an overview of the current state-of-the-art on OCR and the different applications of this technology are commented.

1.2.2 Handwritten Text Recognition

HTR can be considered a relatively new field on PR. An overview about the state-of-the-art can be found on [KGS99, PS00, Vin02]. On [Bun03] a detailed description of what has been made on HTR, what is being made nowadays and the most possible progress on a near future is given.

According to the mode of data acquisition used, automatic handwriting recognition systems can be classified into *on-line* and *off-line*. In *Off-line* systems the handwriting is given as an image or scanned text, without time sequence information. In *On-line* systems the handwriting is given as a temporal sequence of coordinates that represents the pen tip trajectory. In this thesis the main system (see Chapter 4) is an *off-line* system. However, the feedback provided by the user in the MM-CATTI (see Chapter 5) comes in the form of *on-line* data.

Current off-line handwriting transcription products rely on technology for isolated character recognition (OCR) developed in the last decade. First, the input image is segmented into characters or pieces of a character. Thereafter, an isolated character recognition is performed on these segments. On [BS89, KFK02, SR98] this approximation is followed, using some segmentation techniques based on dynamic programming. As it was previously explained, the difficulty of this approach is the segmentation step. In fact, the difficulty of character segmentation for some handwritten documents, makes impossible to apply this approach to this kind of documents.

The required technology should be able to recognize all text elements (sentences, words and characters) as a whole, without any prior segmentation of the image into these elements. This technology is generally referred to as segmentation-free “*off-line Handwritten Text Recognition*” (HTR). HMM technology is widely used in segmentation-free recognizers [BSM99, PS00, MB01, KSS03, TJV04, RAB07].

However, the results obtained with these systems are still very far from offering usable accuracy. They have proven to be suited for restricted applications with very limited vocabulary or constrained handwriting achieving in these kind of tasks relatively high recognition rates. However, for (high quality, modern) unrestricted text images, current HTR state-of-the-art (research) prototypes provide accuracy levels that range from 40 to 80% at the word level. This fact has lead us to propose the development of computer-assisted solutions based on novel approaches.

On the other hand, current on-line handwriting transcription products are more accurate than off-line systems [MFW, PLG01, JMRW01], reaching 90% word-level accuracy in some cases.

1.3 Interactive Pattern Recognition

The idea of interaction between humans and machines is by no means new. In fact, historically, machines have mostly been developed with the aim of assisting human beings in their work. Since the introduction of computer machinery, however, the idea of fully automatic devices that would completely substitute the humans in certain types of tasks, has been gaining increasing popularity. This is the case in areas such as PR. Scientific and technical research in this area have followed the “full automation” paradigm traditionally, even though, in practice, full automation often proves elusive or unnatural in many applications where technology is expected to *assist* rather than replace the human agents.

Placing PR within the human-interaction framework requires fundamental changes in the way we look at problems in this area. Interestingly, these changes entail important research opportunities which hold promise for a new generation of truly human-friendly PR devices. Some ideas within this Interactive Pattern Recognition (IPR) framework were presented in [VRCGV07], where tight interrelations between *user feedback*, *multimodality* and *adaptive learning* are examined.

The CATTI and MM-CATTI scenarios presented in this work (Chapter 4 and Chapter 5 respectively) are based on these ideas. Following these ideas we review in this section how human feedback can be directly used to improve the system performance and discuss the multimodal issues entailed by the resulting IPR framework. Since adaptive learning is not

considered in this thesis, system operation is supposed to be driven by *fixed* statistical models.

In traditional PR [DH73], for a given input x , a best hypothesis is one which maximizes the posterior probability:

$$\hat{h} = \operatorname{argmax}_{h \in \mathcal{H}} \Pr(h | x) \quad (1.1)$$

where \mathcal{H} is the set of possible hypotheses. Now, interaction allows adding more *conditions*:

$$\hat{h} = \operatorname{argmax}_{h \in \mathcal{H}} \Pr(h | x, f) \quad (1.2)$$

where f stands for the feedback, interaction-derived informations; e.g., in the form of *partial hypotheses* or *constraints* on \mathcal{H} . The new system hypothesis, \hat{h} , may prompt the user to provide further feedback informations, thereby starting a new interaction step. The process continues this way until the system output is acceptable by the user. Clearly, the richer the feedback signals f in (1.2) the greater the opportunity to obtain better \hat{h} . But solving the maximization (1.2) may be more difficult than in the case of our familiar $\Pr(h | x)$. Adequate solutions are discussed in the following chapters for HTR applications (see Chapter 4 and Chapter 5).

It is interesting to note that, in general, the interaction feedback informations f do not naturally belong to the original domain from which the main data, x , come from. This entails some sort of *multimodality*. Therefore, eq. (1.2) corresponds to a fairly conventional *modality fusion* problem which can be straight-forwardly re-written as:

$$\hat{h} = \operatorname{argmax}_{h \in \mathcal{H}} \Pr(x, f | h) \cdot \Pr(h) \quad (1.3)$$

In many applications it is natural and/or convenient to assume independence of x and f given h . Consider for instance that x is an image and f the acoustic signal of a speech command possibly describing the image contents. In this case, a *naïve* Bayes decomposition leads to:

$$\begin{aligned} \hat{h} &= \operatorname{argmax}_{h \in \mathcal{H}} \Pr(x | h) \cdot \Pr(f | h) \cdot \Pr(h) \\ &\approx \operatorname{argmax}_{h \in \mathcal{H}} P_{M_X}(x | h) \cdot P_{M_F}(f | h) \cdot P_{M_H}(h) \end{aligned} \quad (1.4)$$

which allows for a separate estimation of independent models, M_X , M_F and M_H for the image and speech components, and the hypothesis prior, respectively^a. As will be discussed later in this thesis, the maximization in eq. (1.4) can often be approached by means of adequate extensions of techniques available to solve the corresponding search problems in the traditional non interactive/multimodal framework.

1.4 Theoretical Background

This section gives an overview of the theoretical foundation of the recognition process. Here the theoretical bases, formulations and main algorithms related with HMMs and n -grams are revised. These statistical models are the basis of the systems developed on this thesis. Finally, the word-graphs technology, used in the CATTI and MM-CATTI systems, is introduced.

^a“True” probabilities are written as $\Pr(\dots)$, in contrast with model approximations such as $P_{M_Z}(z | \dots)$ which, to simplify notation, will be denoted as $P(z | \dots)$ whenever M_z can be understood.

1.4.1 Hidden Markov Models

The Hidden Markov Model (HMM) is a finite set of states, each of which is associated with a continuous (generally multidimensional) probability distribution of “observations”. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution. Only the outcomes, not the states are visible to an external observer and therefore states are “hidden” to the outside; hence the name Hidden Markov Model.

During the past decades it has become the most successful model used in Automatic Speech Recognition (ASR). The main reason for this success is its wonderful ability to characterize the speech signal in a mathematically tractable way. In ASR, HMM observations are discrete time sequences of acoustic parameter vectors. Given the similarity between ASR and HTR, the HMMs have seen increased their popularity in the HTR tasks. In HTR, the HMM observations are also discrete time sequences. However, in this case, the observations represent line-image features and point coordinates of handwritten pen strokes.

HMMs can be classified according to the nature of the observations. When the observations are vectors of symbols in a finite alphabet we are speaking of discrete HMMs. Another possibility is work with continuous observations; the HMMs used in this case are called continuous HMMs. Finally, the third class is called semi-continuous HMMs. These models use discrete observations, but they are modelled using continuous probability density functions [Jel98, Lee89] .

Since in this thesis we work with continuous HMMs, the formal definition and the formulation related with this kind of HMMs is summarized on the next subsections.

Continuous HMM

Here, a formal definition of a continuous HMM is given, using similar notation presented in [Tos04]. We assume that the observations can be only generated at states and not in the transitions. Moreover, an additional initial state, which do not emit any observation, has been defined, in a similar way as in the case of the end state.

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

- Q is a finite set of states. In order to avoid confusions with the indexation of the different states, we are going to call the states of the model as $q_0 \dots q_{|Q|-1}$, whereas the sequence of states that generates the vector sequence \mathbf{x} will be denoted as $z_1 z_2 \dots z_T$.
- I is the initial state, an element of Q : $I \in Q$. $I = q_0$
- F is the final state, an element of Q : $F \in Q$. $F = q_{|Q|-1}$
- X is the real d -dimensional space of observations: $X \subseteq \mathbb{R}^d$.
- a is the state-transition probability function: ^b

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

^b $z_t = q_i$ means that the HMM is on the state q_i at moment t

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

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

- b is a probability distribution function: ^c

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

The following stochastic constraints must be satisfied: $b(q_i, \vec{x}) \geq 0$ and

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

As the observations are continuous then we will have to use a continuous probability density function. In this case probability density function is defined as a weighted sum of G Gaussian distributions:

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

where,

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

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

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

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

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

$$P(z_{t+1} | z_1 \dots z_t) = P(z_{t+1} | z_t)$$

It is called the Markov assumption and the resulting model becomes actually a first order HMM.

^c $x_t = \vec{x}$ means that the HMM on the state z_t gives out \vec{x} at moment t

2. **The stationarity assumption.** Here it is assumed that state transition probabilities are independent of the actual time at which the transitions takes place. Mathematically,

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

for any t_1 and t_2

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

$$P(x_t | x_1 \dots x_{t-1}, z_1 \dots z_t) = P(x_t | z_t)$$

Basic algorithms for HMMs

Once we have an HMM, there are three problems of interest. The evaluation problem, the decoding problem and the learning problem.

- **The Evaluation Problem.** This problem consist on computing the probability $P(\mathbf{x}|M)$. Given an HMM M and a sequence of observations $\mathbf{x} = (\vec{x}_1 \vec{x}_2 \dots \vec{x}_T)$ with $\vec{x}_i \in \mathfrak{R}^d$, this is, the probability that the observations are generated by the model.
- **The Decoding Problem.** Given a model M and a sequence of observations \mathbf{x} , the problem consist on find the most likely state sequence in the model that produced the observations. In other words, the problem consists on find the hidden part of the HMM.
- **The Learning Problem.** Given a model M and a sequence of observations \mathbf{x} , how should we adjust the model parameters M in order to maximize the probability $P(\mathbf{x}|M)$.

To simplify the notation, in the next sections, $a(q_i, q_j)$ will be written as a_{ij} , $b(q_i, x)$ as $b_i(x)$ and the observation of the sequence \mathbf{x} at the moment t , \vec{x}_t , will be written as x_t ^d.

The Evaluation Problem and the Forward and Backward Algorithms

Let $\mathbf{x} = (x_1 x_2 \dots x_T)$ with $x_i \in \mathfrak{R}^d$ a sequence of real vectors and $Z = \{\mathbf{z} = (z_1 z_2 \dots z_T) : z_k = q_i \in (Q - \{I, F\}), 1 \leq i \leq |Q| - 2\}$ a set of state sequences associated with the vector sequence \mathbf{x} . Then, the probability that \mathbf{x} be generated by the model M is:

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

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

This calculation involves number of operations in the order of N^T , where N is the number of states of the model excluding the initial state, $N = |Q| - 1$ ($Q = \{q_0 = I, q_1, \dots, q_{N-1}, q_N = F\}$), and T is the number of vectors on the sequence. This is very large even if the

^dFrom now on, any kind of sequence will be represented as $l_i \dots l_j$ or as \mathbf{l}_i^j , according to convenience.

length of the sequence, T is moderate. Therefore we have to look for an other method for this calculation.

The **Forward** algorithm is an efficient algorithm to compute $P(\mathbf{x}|M)$. The time complexity order of this algorithm is: $O(|Q|^2 \cdot T)$; however, using a left-to-right HMM the complexity falls to $O(|Q| \cdot T)$. In this topology a transition between two states $q_i, q_j \in Q$ from the HMM, it is only possible if $j \geq i$.

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

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

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

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

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

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

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

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

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

The Decoding Problem and the Viterbi Algorithm

In this case we want to find the most likely state sequence, $\mathbf{z} = (z_1 z_2 \dots z_T)$, of the model M , for a given sequence of observations, $\mathbf{x} = (x_1 x_2 \dots x_T)$. The algorithm used here is commonly known as the Viterbi algorithm. This algorithm is similar to the forward algorithm, but replacing the sum by the dominating term.

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

with the condition that $v_0(1) = 1$. The probability of the sequence \mathbf{x} to be emitted by the model M is computed as:

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

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

The Learning Problem and the Baum-Welch Algorithm

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

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

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

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

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

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

where

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

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

$$\begin{aligned} \hat{\mu}_{jg} &= \frac{\sum_{r=1}^R \sum_{t=1}^{T_r} L_{jg}^r(t) x_{rt}}{\sum_{r=1}^R \sum_{t=1}^{T_r} L_{jg}^r(t)} \\ \hat{\Sigma}_{jg} &= \frac{\sum_{r=1}^R \sum_{t=1}^{T_r} L_{jg}^r(t) (x_{rt} - \hat{\mu}_{jg})(x_{rt} - \hat{\mu}_{jg})'}{\sum_{r=1}^R \sum_{t=1}^{T_r} L_{jg}^r(t)} \\ c_{jg} &= \frac{\sum_{r=1}^R \sum_{t=1}^{T_r} L_{jg}^r(t)}{\sum_{r=1}^R \sum_{t=1}^{T_r} L_j^r(t)} \end{aligned}$$

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

Sometimes, it is necessary to have a composition of C HMM joined sequentially, for example in the case of the different letters that conform a sentence. In this case, the “embedded training Baum-Welch” algorithm, which re-estimates the parameters of the composition of C sequentially concatenated HMMs, can be used. This algorithm provides us the possibility to train the HMM without any prior segmentation of the training images into word or characters. On [Tos04] we can find all the formulas to compute the unknown parameters in this case.

1.4.2 Language models: N -grams

Language models (LMs) are used to model text properties like syntax and semantic independently from the morphological models. They are used in many natural language processing applications such as speech recognition, machine translation or handwritten recognition. These models try to capture the properties of a language, and are used to predict the next word in a word sequence. Language models assign a probability to a sequence of l words $\mathbf{w} = w_1, w_2, \dots, w_l$, which can be expressed as:

$$\Pr(\mathbf{w}) = \Pr(w_1) \cdot \prod_{i=2}^l \Pr(w_i | \mathbf{w}_1^{i-1})$$

where $\Pr(w_i | \mathbf{w}_1^{i-1})$ is the probability of the word w_i when we have already seen the sequence of words $w_1 \dots w_{i-1}$. The sequence of words prior to w_i is called history.

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

An n -gram defines a function: $\Phi_n : V^* \rightarrow V^{n-1}$ in which, all sequences finishing with the same $n-1$ words belong to the same equivalence class. Now, $\Pr(\mathbf{w})$ can be approximated as:

$$\Pr(\mathbf{w}) \approx \prod_{i=1}^l P(w_i | \Phi_n(\mathbf{w}_1^{i-1})) = \prod_{i=1}^l P(w_i | \mathbf{w}_{i-n+1}^{i-1}) \quad (1.5)$$

Owing to the fact that, for the first $n-1$ words in \mathbf{w} , $i-n \leq 0$, the Equation (1.5) must be written as:

$$\Pr(\mathbf{w}) \approx P(w_1) \cdot \prod_{i=2}^{n-1} P(w_i | \mathbf{w}_1^{i-1}) \cdot \prod_{i=n}^l P(w_i | \mathbf{w}_{i-n+1}^{i-1}) \quad (1.6)$$

Given a vocabulary V and a transcribed training data or text corpora represented by $\mathbf{w} = w_1 w_2 \dots w_l$, the estimated probability of the word $v \in V$, having seen a sequence of $n-1$

words $\mathbf{v} \in V^{n-1}$, is computed as:

$$P(v|\mathbf{v}) = \frac{C(\mathbf{v}v)}{C(\mathbf{v})}$$

where $C(\mathbf{v})$ is the number of times that the sequence \mathbf{v} has appeared on the training sequence \mathbf{w} . This is a maximum likelihood (ML) estimate.

Since not all possible n -grams have typically been seen in training, some smoothing method must be used to allow for unseen n -grams in the recognition phase. Two main smoothing techniques are: interpolation [Jel98] and ‘‘Back-off’’ [Kat87]. In this thesis we are going to pay attention to the ‘‘Back-off’’ method.

The method presented by Katz consists in discounting a small mass of probability from the seen events, and to distribute it between the unseen events, or those that have been seen very few times, using a $(n-1)$ -gram model also smoothed by ‘‘back-off’’. This is a recursive function that can be expressed as:

$$\hat{P}_{bo}(v_i|\mathbf{v}_{i-n+1}^{i-1}) = \begin{cases} f(v_i|\mathbf{v}_{i-n+1}^{i-1}) & \text{if } C(\mathbf{v}_{i-n+1}^i) > k \\ \alpha(\mathbf{v}_{i-n+1}^{i-1})\hat{P}_{bo}(\mathbf{v}_{i-n+2}^{i-1}) & \text{otherwise} \end{cases}$$

where $f(v_i|\mathbf{v}_{i-n+1}^{i-1})$ is a discounting function that reserve some probability mass for the unseen events and $\alpha(\mathbf{v}_{i-n+1}^{i-1})$ ensures that the model will be consistent.

Essentially, this function means that if the n -gram has been seen k or more times in training, the conditional probability of a word given its history is proportional to the maximum likelihood estimate of that n -gram. Otherwise, the conditional probability is equal to the back-off conditional probability of the $(n-1)$ -gram. The value of k is usually chosen to be 0. However, empirical testing may find better values for k .

There are a lot of successful discount techniques as, for example, Good-Turing [Kat87], Witten-Bell [WB91], or the linear and absolute discounting methods [HMK94]. In this work only the Kneser-Ney discount, presented on [KN95], will be used.

In the Kneser-Ney discounting, the discounted probability is computed by subtracting a constant D from the n -gram count. The main idea of Kneser-Ney is to use a modified probability estimate for lower order n -grams used for backoff. Specifically, the modified probability for a lower order n -gram is taken to be proportional to the number of unique words that precede it in the training data. We can define the discounted probability as:

$$f(v_i|\mathbf{v}_{i-n+1}^{i-1}) = \frac{C(\mathbf{v}_{i-n+1}^i) - D0}{C(\mathbf{v}_{i-n+1}^{i-1})} \quad \text{for highest order n-grams}$$

$$f(v_i) = \frac{n(*v_i) - D1}{n(**)} \quad \text{for lower order n-grams}$$

where $n(*v_i) = |\{v_{i-1} : C(v_{i-1}v_i) > 0\}|$ is the number of different words v_{i-1} that precede v_i in the training data and where $n(**) = |\{(v_{i-1}, v_i) : C(v_{i-1}v_i) > 0\}| = \sum_{v_i} n(*v_i)$. $D0$ and $D1$ represents two different discounting constants, because the original kneser-ney discounting uses one discounting constant for each n -gram order. These constants are estimated

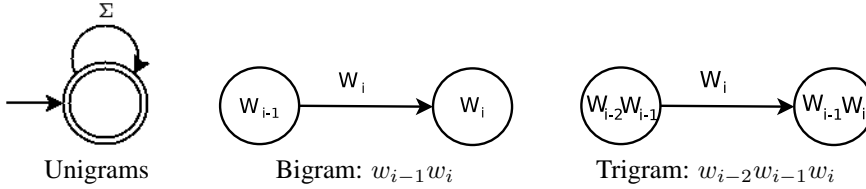


Figure 1.3: Examples of n -grams represented using a SFSA.

as:

$$D = \frac{n1}{(n1 + 2 \cdot n2)}$$

where $n1$ and $n2$ are the total number of n -grams with exactly one and two counts

n -grams modelled by a stochastic finite state automaton

Along this work, is very common use stochastic finite state automation (SFSA) to represent HMMs, lexical models and language models. Thanks to this homogeneous finite-state (FS) nature of all these models, they can be easily integrated into a single global FS model. The n -grams can be represented using a SFSA [VTC+05a, VTC+05b]. It is represented using the sextuple $A = (Q, V, \delta, q_0, P, F)$, where:

- V is non-empty finite set of symbols
- $Q \subseteq V^{n-1} \cup q_0$ is a finite, not-empty set of states. Each state is defined using the vocabulary symbols V as $q = (v_{i-n+1} \dots v_{i-2} v_{i-1}) \in Q$
- $\delta \subset Q \times V \times Q$ is the state-transition function. The transitions is :

$$(v_{i-n+1} \dots v_{i-2} v_{i-1}, v, v_{i-n+2} \dots v_{i-1} v)$$

where $(v_{i-n+1} \dots v_{i-2} v_{i-1}) \in Q$, $(v_{i-n+2} \dots v_{i-1} v) \in Q$, and $v \in V$

- q_0 is the initial state ($q_0 \in Q$)
- $P : \delta \rightarrow \mathfrak{R}^+$ is the probability transition function. We are using deterministic SFSA, so each transition is identified with only the source state $q \in V^{n-1}$ and the transition symbol $v \in V$. Therefore, $P(q, v, q') = P(v|q)$
- $F : Q \rightarrow \mathfrak{R}^+$ is the final state probability function.

Figure 1.3 shows examples of SFSA to represent 1-gram, 2-grams and 3-grams respectively. On Figure 1.4 we can see the example of a back-off smoothed n -gram presented using SFSA.

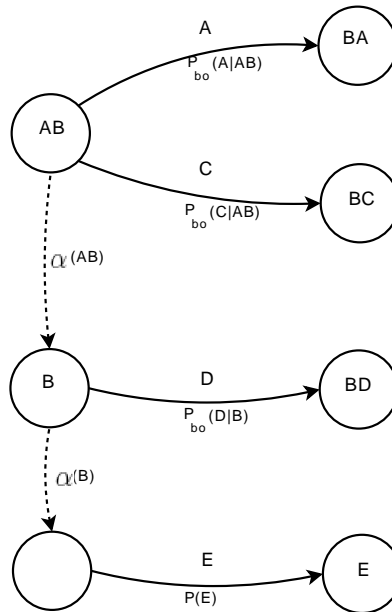


Figure 1.4: Examples of a SFSA representing a back-off smoothed n -gram.

1.4.3 Word-graphs

As we will see on Chapter 4, a direct adaptation of the Viterbi algorithm to implement the CATTI system leads to a computational cost that grows quadratically with the number of words of each sentence. This can be problematic for large sentences and/or for fine-grained (character-level) interaction schemes. In this work word-graph techniques that can achieve very efficient, linear cost search have been studied. In this section the basic word graph concepts are introduced.

A word graph (WG) is a data structure that represents a finite sample of word sequences in a very efficient way. Formally, a WG is represented as a weighted directed acyclic graph (WDAG) defined by the eight tuple $WG = (Q, V, A, n_I, F, t, p, \omega)$:

- Q is a finite set of nodes. Since a WG is a DAG a topological order on the nodes is assumed. Each node n is labelled with its corresponding index in this order: $Q = \{1, 2, \dots, |Q|\}$.
- V is a non-empty set of words (vocabulary).
- $A \subset Q \times Q$ is a finite set of edges. Each edge e is denoted by its start and end nodes: $e = (i, j)$ where $i \in Q, j \in Q$ and $i < j$.
- $n_I \in Q$ is the initial node: $n_I = 1$.
- $F \subseteq Q$ is the set of final nodes.

- $t : Q \rightarrow \{1\dots T\}$ is a function that associates each node with a point (horizontal position) of the handwritten image. Typically $t(n_I) = 1$ and $\forall_{n \in F} t(n) = T$.
- $\omega : A \rightarrow V$ is a function that relates each edge with a word. Typically, given the node $e = (i, j)$, the related word $\omega(e)$ is a word hypothesis between horizontal image positions $t(i)$ and $t(j)$.
- $p : A \rightarrow [0, 1]$ is an edge probability function. Typically, given an edge $e = (i, j)$, $p(e)$ is the probability of the hypothesis that $\omega(e)$ appears between $t(i)$ and $t(j)$.

An example of a word graph is shown in Figure 1.5. This word graph represents a set of possible transcriptions of the handwritten Spanish sentence “*antiguos ciudadanos que en Castilla se llamaban*”.

The word sequences represented by the WG are formed by words on paths in the WG from the initial node to a final node. Formally, given a path $\phi = \{e_1 = (1, i_1), e_2 = (i_1, i_2), \dots, e_l = (i_{l-1}, i_l)\}$ on WG , where $i_l \in F$, the word sequence associated to this path is $\mathbf{w} = \omega(e_1), \omega(e_2), \dots, \omega(e_l)$ and its probability is computed as the product of the probabilities of the edges along the path:

$$P(\phi) = \prod_{i=1}^l p(e_i) \quad (1.7)$$

However, given that the WG is ambiguous (for each node and word pair there may be several possible next nodes), in general there is more than one path that generates the sequence \mathbf{w} . If $d(\mathbf{w})$ is defined as the set of all the paths associated with \mathbf{w} and $\phi_{\mathbf{w}}$ is one of these paths, the probability of the word sequence \mathbf{w} is computed as:

$$P(\mathbf{w}) = \sum_{\phi_{\mathbf{w}} \in d(\mathbf{w})} P(\phi_{\mathbf{w}}) \quad (1.8)$$

Given a WG, the word sequence with greater probability can be written as:

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} \sum_{\phi_{\mathbf{w}} \in d(\mathbf{w})} P(\phi_{\mathbf{w}}) \quad (1.9)$$

It should be noted that the maximization problem stated in Equation (1.9) is NP-hard [CH00]. Nevertheless, adequate approximations can be obtained by means of efficient search algorithms, like Viterbi [Vit67]:

$$P(\mathbf{w}) \approx P(\tilde{\mathbf{w}}) = \max_{\phi_{\mathbf{w}} \in d(\mathbf{w})} P(\phi_{\mathbf{w}}) \quad (1.10)$$

$$\hat{\mathbf{w}} \approx \tilde{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} \max_{\phi_{\mathbf{w}} \in d(\mathbf{w})} P(\phi_{\mathbf{w}}) \quad (1.11)$$

For example, given the WG represented in Figure 1.5, the probability of the path h_1 :

$$h_1 = \{(1, 2), (2, 6), (6, 8), (8, 11), (11, 12), (12, 13), (13, 15), (15, 18)\}$$

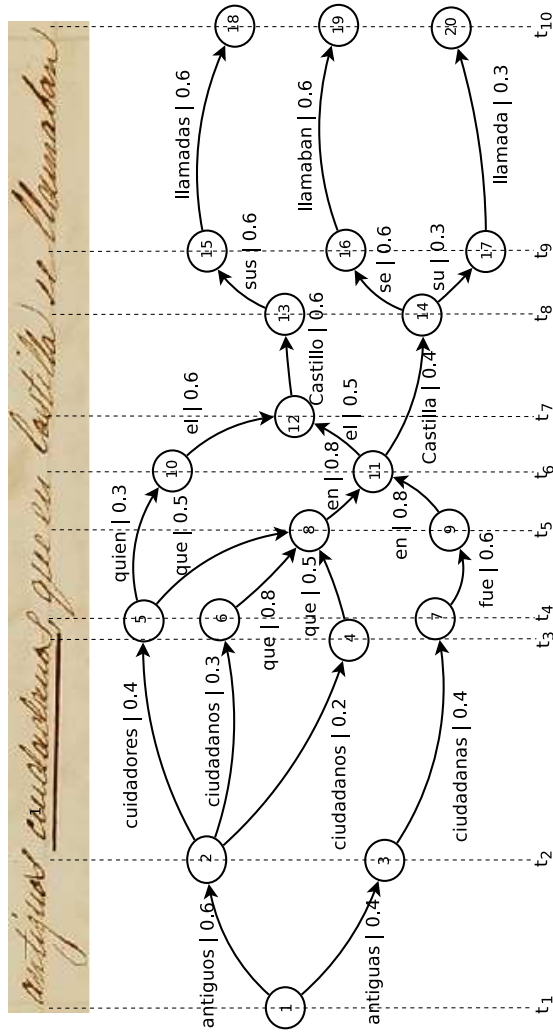


Figure 1.5: Image positions t_i are associates with the nodes as: $t(1) = t_1$, $t(2) = t(3) = t_2$, $t(4) = t_3$, $t(5) = t(6) = t(7) = t_4$, $t(8) = t(9) = t_5$, $t(10) = t(11) = t_6$, $t(12) = t_7$, $t(13) = t(14) = t_8$, $t(15) = t(16) = t(17) = t_9$, $t(18) = t(19) = t(20) = t_{10}$

is computed as:

$$\begin{aligned} P(h_1) &= p(1,2)p(2,6)p(6,8)p(8,11)p(11,12)p(12,13)p(13,15)p(15,18) \\ &= 0.6 \cdot 0.3 \cdot 0.8 \cdot 0.8 \cdot 0.5 \cdot 0.6 \cdot 0.6 \cdot 0.6 = 0.012 \end{aligned}$$

The word sequence associated with this path is \mathbf{w}_1 ="antiguos ciudadanos que en el Castillo sus llamadas". However, h_1 is not the only path that generates \mathbf{w}_1 ,

$$h_2 = \{(1, 2), (2, 4), (4, 8), (8, 11), (11, 12), (12, 13), (13, 15), (15, 18)\}$$

generates it too. So, the exact probability of \mathbf{w}_1 is:

$$P(\mathbf{w}_1) = P(h_1) + P(h_2) = 0.012 + 0.005 = 0.017$$

and approximating by the Viterbi algorithm, the probability of \mathbf{w}_1 is the probability of the path with maximum probability, i.e:

$$P(\mathbf{w}_1) \approx P(\tilde{\mathbf{w}}_1) = P(h_1) = 0.012$$

Now, to obtain the word sequence with greater probability all the word sequences on the WG must be taken into account. Next, we can see all the word sequences on the WG in Figure 1.5 and its corresponding paths:

- \mathbf{w}_1 ="antiguos ciudadanos que en el Castillo sus llamadas"
 $h_1 = \{(1, 2), (2, 6), (6, 8), (8, 11), (11, 12), (12, 13), (13, 15), (15, 18)\}$
 $h_2 = \{(1, 2), (2, 4), (4, 8), (8, 11), (11, 12), (12, 13), (13, 15), (15, 18)\}$
- \mathbf{w}_2 ="antiguos ciudadanos que en Castilla se llamaban"
 $h_3 = \{(1, 2), (2, 6), (6, 8), (8, 11), (11, 14), (14, 16), (16, 19)\}$
 $h_4 = \{(1, 2), (2, 4), (4, 8), (8, 11), (11, 14), (14, 16), (16, 19)\}$
- \mathbf{w}_3 ="antiguos ciudadanos que en Castilla su llamada"
 $h_5 = \{(1, 2), (2, 6), (6, 8), (8, 11), (11, 14), (14, 17), (17, 20)\}$
 $h_6 = \{(1, 2), (2, 4), (4, 8), (8, 11), (11, 14), (14, 17), (17, 20)\}$
- \mathbf{w}_4 ="antiguos cuidadores quien el Castillo sus llamadas"
 $h_7 = \{(1, 2), (2, 5), (5, 10), (10, 12), (12, 13), (13, 15), (15, 18)\}$
- \mathbf{w}_5 ="antiguos cuidadores que en el Castillo sus llamadas"
 $h_8 = \{(1, 2), (2, 5), (5, 8), (8, 11), (11, 12), (12, 13), (13, 15), (15, 18)\}$
- \mathbf{w}_6 ="antiguos cuidadores que en Castilla se llamaban"
 $h_9 = \{(1, 2), (2, 5), (5, 8), (8, 11), (11, 14), (14, 16), (16, 19)\}$
- \mathbf{w}_7 ="antiguos cuidadores que en Castilla su llamada"
 $h_{10} = \{(1, 2), (2, 5), (5, 8), (8, 11), (11, 14), (14, 17), (17, 20)\}$
- \mathbf{w}_8 ="antiguas ciudadanas fue en el Castillo sus llamadas"
 $h_{11} = \{(1, 3), (3, 7), (7, 9), (9, 11), (11, 12), (12, 13), (13, 15), (15, 18)\}$

- \mathbf{w}_9 = “antiguas ciudadanas fue en Castilla se llamaban”
 $h_{12} = \{(1, 3), (3, 7), (7, 9), (9, 11), (11, 14), (14, 16), (16, 19)\}$
- \mathbf{w}_{10} = “antiguas ciudadanas fue en Castilla su llamada”
 $h_{13} = \{(1, 3), (3, 7), (7, 9), (9, 11), (11, 14), (14, 17), (17, 20)\}$

and the probabilities are:

$$\begin{aligned}
 P(\mathbf{w}_1) &= P(h_1) + P(h_2) = 0.012 + 0.005 = 0.017 & P(\mathbf{w}_6) &= P(h_9) = 0.014 \\
 P(\mathbf{w}_2) &= P(h_3) + P(h_4) = 0.016 + 0.007 = 0.023 & P(\mathbf{w}_7) &= P(h_{10}) = 0.003 \\
 P(\mathbf{w}_3) &= P(h_5) + P(h_6) = 0.004 + 0.002 = 0.006 & P(\mathbf{w}_8) &= P(h_{11}) = 0.008 \\
 P(\mathbf{w}_4) &= P(h_7) = 0.009 & P(\mathbf{w}_9) &= P(h_{12}) = 0.011 \\
 P(\mathbf{w}_5) &= P(h_8) = 0.010 & P(\mathbf{w}_{10}) &= P(h_{13}) = 0.003
 \end{aligned}$$

So, the word sequence with greater probability is \mathbf{w}_2 for both exact probabilities ($P(\mathbf{w}_2) = 0.023$) and probabilities approximated using the Viterbi algorithm ($P(\hat{\mathbf{w}}_2) = 0.016$).

Sometimes, it can be necessary not only to compute the best word sequences, but also the n -best word sequences in the word graph. In this work we are going to use the algorithm known as “Recursive Enumeration Algorithm” (REA) [Jm99]. The main reason that support this decision is its simplicity to calculate best paths on demand.

1.5 Scientific objectives

The main objective of this thesis is to put forward the theoretical framework, as well as to develop and to assess a *multimodal computer assisted transcription system for handwritten text images*. More precisely, the scientific objectives of this thesis can be summarized in the next points:

1. Put forward the theoretical framework and develop a computer assisted transcription of handwritten text images system. The intention of this study is to test the interactive-predictive framework, previously applied to machine translation and speech transcription, on the handwritten text images recognition problem.
2. The user is repeatedly interacting with the computer assisted transcription of handwritten text images system. So, one of the objectives of this thesis is to make the interaction process friendly and ergonomic to the user. Different ways to interact with the system, using the keyboard and the mouse, and different interaction levels (at whole word and at character-keystroke) will be studied in order to improve the interaction process.
3. The response time is a critical feature of the system. The system must be able to interact with the human expert in a time efficient way, otherwise the user will prefer to transcribe the document without any kind of help. In this work we will study different implementations trying to obtain a good response time without loosing too much accuracy.

4. To introduce more ergonomic multimodal interfaces should result in an easier and more comfortable human-machine interaction. In this thesis we intend to develop and to assess a multimodal computer assisted transcription system which focuses on electronic pen or touchscreen communication. These are perhaps the most natural modality to provide the required feedback in the handwritten text image recognition systems here considered.
5. Finally, a demonstrator of the multimodal computer assisted transcription system studied here will be developed. In this way the theoretical framework studied in this work could be fully tested in practice.

As we will see in Chapter 7, all these objectives have been fulfilled to a great extent. In particular, the multimodal interactive predictive framework has been successfully tested in the handwritten text images recognition problem. A system for Multimodal Computer Assisted Transcription of Handwritten Text Images has been developed based on HMMs and n -gram language models. The system has been automatically tested and, the results suggest that, using the interactive approach, considerable amounts of user effort can be saved with respect to non-interactive HTR systems. Different ways to interact with the system and different levels have been studied improving the system ergonomics and reducing the number of user corrections needed to obtain perfect transcriptions. Finally, word-graph techniques have been studied, obtaining a very efficient, linear cost, which allows the user interact with the system in a time efficient way.

Bibliography

- [AGJ10] Ihab Khoury Adrià Giménez and Alfons Juan. Windowed Bernoulli Mixture HMMs for Arabic Handwritten Word Recognition. In *Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Kolkata (India), Nov 2010 2010.
- [BB08] R. Bertolami and H. Bunke. Hidden markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, 41(11):3452–3460, 2008.
- [BBC+09] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda H. Ney, J. Tomás, and E. Vidal. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.
- [BRST95] H. Bunke, M. Roth, and E.G. Suchakat-Talamazzini. Off-line cursive handwriting recognition using hidden markov models. *Pattern Recognition*, 28(9):1399–1413, 1995.
- [BS89] R. Bozinovic and S. Srihari. Off-line cursive script word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):68–83, 1989.
- [BSM99] I. Bazzi, R. Schwartz, and J. Makhoul. An Omnifont Open-Vocabulary OCR System for English and Arabic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):495–504, 1999.
- [Bun03] H. Bunke. Recognition of cursive roman handwriting- past, present and future. In *Proceedings of the 7th International Conference Document Analysis and Recognition (ICDAR 03)*, page 448, Washington, DC, USA, 2003. IEEE Computer Society.
- [CCC+09] F. Casacuberta, J. Civera, E. Cubel, A.L. Lagarda, G. Lapalme, E. Macklovitch, and E. Vidal. Human interaction for high quality machine translation. *Communications of the ACM*, 52(10):135–138, 2009.
- [CH00] Francisco Casacuberta and Colin De La Higuera. Computational complexity of problems on probabilistic grammars and transducers. In *ICGI '00: Proceedings of the 5th International Colloquium on Grammatical Inference*, pages 15–24, London, UK, 2000. Springer-Verlag.
- [CVC+04] J. Civera, J.M. Vilar, E. Cubel, A.L. Lagarda, S. Barrachina, F. Casacuberta, E. Vidal, D. Picó, and J. González. A syntactic pattern recognition approach to computer assisted translation. In *Advances in Statistical, Structural and Syntactical Pattern Recognition*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
- [DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. J. Wiley and Sons, 1973.

- [DIMP02] G. Dimauro, S. Impedovo, R. Modugno, and G. Pirlo. A new database for research on bank-check processing. In *8th International Workshop on Frontiers in Handwriting Recognition*, pages 524–528, 2002.
- [FLS92] R. Fenrich, S. Lam, and S. N. Srihari. Chapter optical character recognition. In *Encyclopedia of Computer Science and Engineering*, pages 993–1000. Third ed., Van Nostrand, 1992.
- [HNK94] U. Essen H. Ney and R. Kneser. On structuring probabilistic dependencies in stochastic language modeling. *Computer speech and Language*, 8(1):1–28, 1994.
- [IOO91] S. Impedovo, L. Ottaviano, and S. Occhiegro. Optical character recognition a survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1):1–24, 1991.
- [Jel98] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.
- [Jm99] Víctor M. Jiménez and Andrés marzal. Computing the k shortest paths: a new algorithm and an experimental coparison. In J. S. Viter and C. D. Zaraliagis, editors, *Algorithm Engineering*, volume 1668 of *Lecture Notes in Computer Science (LNCS)*, pages 15–29. Springer-Verlag, July 1999.
- [JMRW01] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. On-Line Handwriting Recognition: The NPen++ Recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–181, 2001.
- [JV01] A. Juan and E. Vidal. On the use of Bernoulli mixture models for text classification. In *Proceedings of the Workshop on Pattern Recognition in Information Systems (PRIS 01)*, Setúbal (Portugal), July 2001.
- [Kat87] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35:400–401, March 1987.
- [KFK02] E. Kavallieratou, N. Fakotakis, and G. Kokkinaki. An unconstrained handwriting recognition system. *International Journal of Document Analysis and Recognition*, 4:226–242, 2002.
- [KGS99] G. Kim, V. Govindaraju, and S. Srihari. An architecture for handwritten text recognition systems. *International Journal of Document Analysis and Recognition*, 2(1):37–44, 1999.
- [KN95] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. volume 1, pages 181–184, Los Alamitos, CA, USA, 1995. IEEE Computer Society.
- [KPNV02] D. Keysers, R. Paredes, H. Ney, and E. Vidal. Combination of tangent vectors and local representations for handwritten digit recognition. In *International*

- Workshop on Statistical Pattern Recognition*, Lecture Notes in Computer Science, pages 407–441. Springer-Verlag, 2002.
- [KSS03] A. L. Koerich, R. Sabourin, and C. Y. Suens. Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis Applications*, 6(2):97–121, 2003.
- [Lee89] K.-F. Lee. Automatic speech recognition: The development of the sphinx system. *Kluwer Academic Publishers, Boston/Dordrecht/London*, 1989.
- [Lee96] S. W. Lee. Off-line recognition of totally unconstrained handwritten numerals using mcn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:648–652, 1996.
- [LS06] P. Liu and F. K. Soong. Word graph based speech recognition error correction by handwriting input. In *ICMI '06: Proceedings of the 8th international conference on Multimodal Interfaces*, pages 339–346, New York, NY, USA, 2006. ACM.
- [MB99] U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *Proceedings of the International Conference Document Analysis and Recognition (ICDAR'99)*, pages 705–708, Bangalore (India), 1999.
- [MB01] U.-V. Marti and H. Bunke. Using a Statistical Language Model to improve the performance of an HMM-Based Cursive Handwriting Recognition System. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.
- [MFW] S. Manke, M. Finke, and A. Waibel. Npen++: A writer independent, large vocabulary on-line cursive handwriting recognition system. In *International Conference on Document Analysis and Recognition*, Montreal.
- [MSY92] S. Mori, C. Y. Suen, and K. Yamamoto. Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.
- [Nag00] G. Nagy. Twenty years of document image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.
- [PLG01] M. Parizeau, A. Lemieux, and C. Gagn. Character Recognition Experiments using Unipen Data. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 481–485, 2001.
- [PS00] R. Plamondon and S. N. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.
- [Rab89] L. Rabiner. A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. *Proceedings IEEE*, 77:257–286, 1989.

- [RAB07] V. Romero, V. Alabau, and J. M. Benedí. Combination of N-grams and Stochastic Context-Free Grammars in an Offline Handwritten Recognition System. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *LNCIS*, pages 467–474. Springer-Verlag, Girona (Spain), June 2007.
- [Rat03] E. H. Ratzlaff. Methods, Report and Survey for the Comparison of Diverse Isolated Character Recognition Results on the UNIPEN Database. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR '03)*, volume 1, pages 623–628, Edinburgh, Scotland, August 2003.
- [RCV07] L. Rodríguez, F. Casacuberta, and E. Vidal. Computer Assisted Speech Transcription. In *Proceedings of the third Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *Lecture Notes in Computer Science*, pages 241–248. Girona (Spain), June 2007.
- [RGJ07] V. Romero, A. Giménez, and A. Juan. Explicit Modelling of Invariances in Bernoulli Mixtures for Binary Images. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *Lecture Notes in Computer Science*, pages 539–546. Springer-Verlag, Girona (Spain), June 2007.
- [SdIfIV+] SchlumbergerSema S.A., Instituto Tecnológico de Informática, Rheinisch Westfälische Technische Hochschule Aachen Lehrstuhl für Informatik VI, Recherche Appliquée en Linguistique informatique Laboratory University of Montreal, Celer Soluciones, Société Gamma, and Xerox Research Centre Europe. In *TT2. TransType2 - computer assisted translation. Project technical annex. information Society Technologies (IST) Programme.*, number IST-2001-32091.
- [She53] D. H. Shepard. Apparatus for reading. *US-Patent No 2663758*, 1953.
- [SK97] S. N. Srihari and E. J. Keubert. Integration of handwritten address interpretation technology into the united states postal service remote computer reader system. In *Fourth International Conference Document Analysis and Recognition*, volume 2, pages 892–896, Ulm, Germany, August 1997.
- [SMW01] B. Suhm, B. Myers, and A. Waibel. Multimodal Error Correction for Speech User Interfaces. *ACM Transactions on Computer-Human Interaction*, 8(1):60–98, March 2001.
- [SR98] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.
- [Sri93] S. Srihari. Recognition of handwritten and machine printed text for postal address interpretation. *Pattern Recognition Letters*, 14:291–302, 1993.

-
- [TJV04] Alejandro H. Toselli, Alfons Juan, and Enrique Vidal. Spontaneous Handwriting Recognition and Classification. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 433–436, Cambridge, United Kingdom, August 2004.
- [Tos04] Alejandro Héctor Toselli. *Reconocimiento de Texto Manuscrito Continuo*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia (Spain), March 2004. Advisor(s): Dr. E. Vidal and Dr. A. Juan (in Spanish).
- [VCR+07] E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer Assisted Translation using speech Recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941–951, 2007.
- [Vin02] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1033–1446, 2002.
- [Vit67] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.
- [VRCGV07] E. Vidal, L. Rodríguez, F. Casacuberta, and I. García-Varea. Interactive pattern recognition. In *Proceedings of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 4892 of *Lecture Notes in Computer Science*, pages 60–71. Brno, Czech Republic, June 2007.
- [VTC+05a] E. Vidal, F. Thollard, F. Casacuberta, C. de la Higuera, and R. Carrasco. Probabilistic finite-state machines - part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025, 2005.
- [VTC+05b] E. Vidal, F. Thollard, F. Casacuberta, C. de la Higuera, and R. Carrasco. Probabilistic finite-state machines - part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1025–1039, 2005.
- [WB91] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transaction on Information Theory*, 17, 1991.

CHAPTER 2

Corpora

2.1 Introduction

In this chapter, the main features of the different corpora that have been used throughout this thesis are exposed. The systems exploited on this work are based on supervised training, therefore not only the different samples are needed, but also the transcriptions of these samples.

We have used four different corpora. Three of them are off-line handwritten text corpora, and the last one is an on-line handwritten text corpus. The first off-line corpus was compiled from an historic handwritten document identified as “Cristo-Salvador” (CS). The second one, called ODEC, consists of handwritten answers from survey forms. And the last off-line corpus, called IAMDB, consists of handwritten full English sentences based on the Lancaster-Oslo/Bergen (LOB) corpus. On the other hand, the on-line corpus, called UNIPEN, is an English dataset divided into several categories: letters, digits, symbols, isolated words and full sentences.

2.2 Cristo Salvador

This corpus was compiled from the legacy handwritten document from the XIX century identified as “Cristo-Salvador”, which was kindly provided by the *Biblioteca Valenciana Digital* (BIVALDI)^a.

^a<http://bv2.gva.es>



Figure 2.1: Examples of the corpus “Cristo-Salvador”.

This was written by only one writer and scanned at 300dpi. This corpus is a legacy document and it suffers the typical degradation problems of this kind of documents [Dri06]. Among these are the presence of smear, significant background variations and uneven illumination, spots due to the humidity, and marks resulting from the ink that goes through the paper (generally called bleed-through). In addition, other kind of difficulties appear in these pages as different sizes in the words, underlined words, etc. The combination of these problems make the recognition of this document a difficult process.

This is a rather small document composed of 53 colour images of text pages. Some of these page images are shown in the Figure 2.1. On Figure 2.2 you can see a detailed portion of one of these pages.

The page images were preprocessed and automatically divided into lines (see Section 3.2.1). The results were visually inspected and the few line-separation errors (around 4%) were manually corrected, resulting in a data-set of 1,172 text line images. The transcriptions corresponding to line images are also available, containing 10,918 running words with a vocabulary of 3,287 different words.

Two different partitions were defined for this data-set. In the first one, called *page* (or *soft*), the test set is formed by 491 line samples corresponding to the last ten lines of each document page, whereas the training set is composed of the 681 remaining lines. On the other hand, in the second partition, called *book* (or *hard*), the test set is composed of 497 line samples belonging to the last 20 document pages, whereas the remaining 675 lines (the 33 initial pages) were assigned to the training set. All the information related with *page* and *book* partitions is summarized in the Tables 2.1 and 2.2 respectively.

The number of words of the test partition that does not appear on the training partition

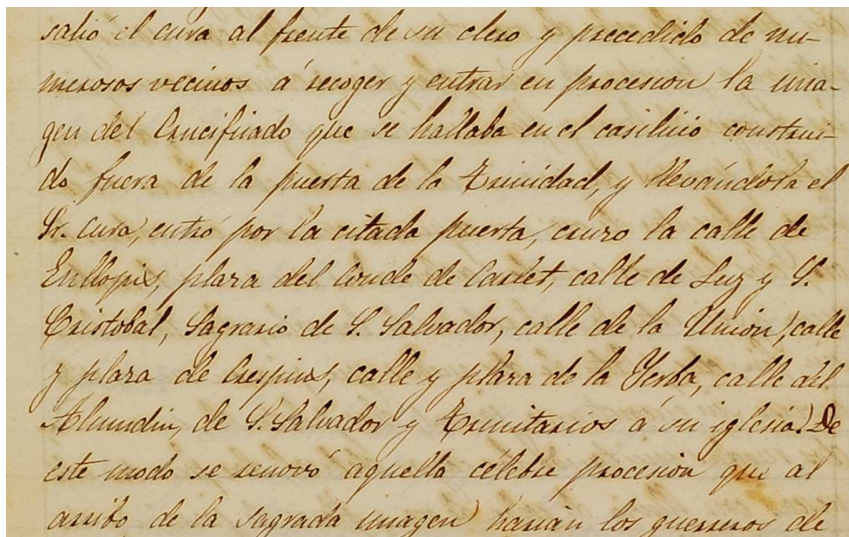


Figure 2.2: Detailed section of a page from the corpus “Cristo-Salvador”.

Table 2.1: Basic statistics of the partition *page* of the database Cristo-Salvador.

Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
Pages	53	53	53	–	–	–
Text lines	681	491	1,172	–	–	–
Words	6,435	4,483	10,918	2,277	1,010	2.8
Characters	36,729	25,487	62,216	78	0	470

Table 2.2: Basic statistics of the partition *book* of the database Cristo-Salvador.

Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
Pages	33	20	53	–	–	–
Text lines	675	497	1,172	–	–	–
Words	6,227	4,691	10,918	2,237	1,050	2.8
Characters	35,863	26,353	62,216	78	0	460

is shown on the column OOV (out of vocabulary words) in the Tables 2.1 and 2.2. On the other hand, the last column shows the “*training ratio*”, that is, the average number of times that one word/character has appeared in the training text.

Note that, the *page* partition is considered to be easier to recognize than the *book* one because its test line samples were extracted from the same pages that the training text line samples. On the other hand, *book* partition better approaches a realistic transcription process. That is, the system is initially trained with the first document pages, and then as more

page images are transcribed, a greater amount of samples (line images and transcriptions) become available to retrain the system and therefore improve the transcription of the rest of the document.

It is important to remark that this corpus has quite a small training ratio (around 2.8 training running words per lexicon-entry). This is expected to result in undertrained (n -gram) language models, which will clearly increase the difficulty of the recognition task.

2.3 ODEC

This corpus entails casual, spontaneous handwriting phrase images [Tos04, TJV04]. It was compiled from handwritten answers extracted from survey forms made for a telecommunication company^b.

The handwritten sentences are answers to the suggestion point of the forms (see Figure 2.3 point 8). These answers were written by a heterogeneous group of people, without any explicit or formal restriction. In addition, since no guidelines were given as to the kind of pen or the writing style to be used, paragraphs become very variable and noisy. Some of the difficulties involved in this task are worth mentioning. In some samples, the stroke thickness is non-uniform. Other samples present irregular and non-consistent spacing between words and characters. Also, there are words written using different case, font types and sizes intra-word and some samples including foreign language words or sentences. On the other hand, noise and non-textual artefacts often appear in the paragraphs. Among these noisy elements appear unknown words or words containing orthographic mistakes, as well as underlined and crossed-out words. Unusual abbreviations and symbols, arrows, etc. are also within this category. The combination of the writing-styles variability and noise may result in partly or entirely illegible samples and make preprocessing operations such as deskew, deslanting, or size normalization, essential. On the Figure 2.4 we can see some examples of this variability.

The images data-set extracted from the ODEC survey forms consists of 913 binary images of handwritten paragraphs scanned at 300 dpi. Because of the inherent difficulty of the task, line extraction was carried out in a semi-automatic way. Most of the paragraphs were processed automatically, but manual supervision was applied to difficult line-overlapping cases such as some of those shown in Figure 2.4. By adequately pasting the extracted lines, a single-line (long) image per form was obtained for each sample (see Figure 2.5). The resulting set of sentences was randomly partitioned into a training set of 676 images and a test set of 237 images. The transcriptions corresponding to the images are also available, containing 16, 371 words with a vocabulary of 3, 308 different words. More information can be found in Table 2.3.

The transcription of sentences was carried out manually, trying to obtain the most detailed transcription possible. So, the words were transcribed in the same way as they appear on the text, with orthographic mistakes, alternating lowercase with uppercase, in the original language, etc. Some codes were defined to label the different things that appear on the text, such as signs, crossed-out words, arrows, etc.

^bData kindly provided by ODEC S.A. (www.odec.es)

1 ¿A través de qué medios le informa habitualmente Telefonía Movistar del servicio telefónico que le ofrece? (Marque con una X tantos como considere)

Publicidad en medios de comunicación Mensajes cortos
 Información recibida por correo Información recibida con la factura

Otros (indique en otro espacio)

2 ¿Se considera bien informado sobre los servicios y novedades de Telefonía Movistar?

Sí No

3 ¿Recuerda haber recibido en su domicilio algún tipo de información sobre el servicio de Telefonía Movistar junto con la factura?

Sí No

Si ha contestado "Sí", ¿puede indicarnos que hace normalmente con esta información?

La lee con atención La mira sin poder entenderla bien
 La tira sin leerla

Si ha marcado "La tira sin leerla", ¿podría señalar el motivo?

No tengo tiempo No me interesa
 No lo recuerdo Otros (conteste en mayúsculas)

4 ¿Recuerda haber recibido en su domicilio algún tipo de comunicación sobre el servicio de Telefonía Movistar dirigida a los usuarios empujados junto con la factura?

Sí No

Si ha contestado "Sí", ¿podría indicarnos que hace normalmente con estas comunicaciones?

Las lee con atención Las mira sin poder entenderla bien
 Las tira sin leerlas

5 Señale su grado de satisfacción general con el servicio de telefonía móvil prestado actualmente. Para responder, utilice una escala de 1 a 10, en la que 1 significa "Nada satisfecho" y 10 "Muy satisfecho"

Nada satisfecho Muy satisfecho

6 ¿Que aspectos del servicio considera que deberían mejorar para que aumentara su satisfacción con el mismo? (conteste en mayúsculas)

EMPLLEAR LAS OPORTUNIDADES DE MERCADO PARA OTRO TIPO DE TARIFAS

7 Si tuviese que recomendar a un amigo o conocido una empresa de telefonía móvil, ¿recomendaría Telefonía Movistar? Utilice la escala de 1 a 10 en la que 1 significa "Nunca lo recomendaré" y 10 "Siempre lo recomendaré"

Nunca lo recomendaré Siempre lo recomendaré

8 Por último, si desea realizar algún comentario o sugerencia sobre el servicio que presta Telefonía Movistar o las comunicaciones que le envía, hágalo a continuación. (conteste en mayúsculas)

QUE FUERA MAS ASQUIBLE ECONOMICAMENTE Y QUE HICIERA UNA PUBLICIDAD MAS SENCILLA

El ejemplo presentado en Opción 15/1999 de la Ley de Protección de Datos de Carácter Personal conforme al art. 5.º f) forma parte de la información en el resultado de la Ley de Telefonía Móvil según el Art. 1.º de la misma. La respuesta al cuestionario se valora. Los clientes que envían el cuestionario completo obtienen 500 puntos del programa de puntos de Movistar Plus. La información aquí contenida es de carácter informativo y no constituye una oferta de servicios. Para más información consulte la información en la página de Internet de Telefonía Móvil. Madrid, 2000. Nadie que sea el destinatario de un mensaje habitualmente por la Red de Telefonía Móvil puede utilizar sus datos para otros fines que no sean los que se indican en el presente. Para cualquier otra información o dudas no compatibles con las presentes condiciones, sustrayendo de ellas que sus contenidos se amparan con la finalidad indicada en el artículo 1.º de la Ley de Protección de Datos de Carácter Personal, dirigirse por escrito a la Ley de Protección de Datos de Carácter Personal, en la calle de Alcalá, 48, 28014 Madrid, España. El titular de los datos podrá ejercitar los derechos de acceso, rectificación, cancelación y oposición previstos en la Ley. Se hace manifestación expresa de que se acojan todos los datos que el cliente no puede oponer en que sus datos serán utilizados por la empresa de Grupo Telefónica exclusivamente para el agente de ventas de los servicios que pueden ser de interés.

Figure 2.3: A sample form from the ODEC Database.

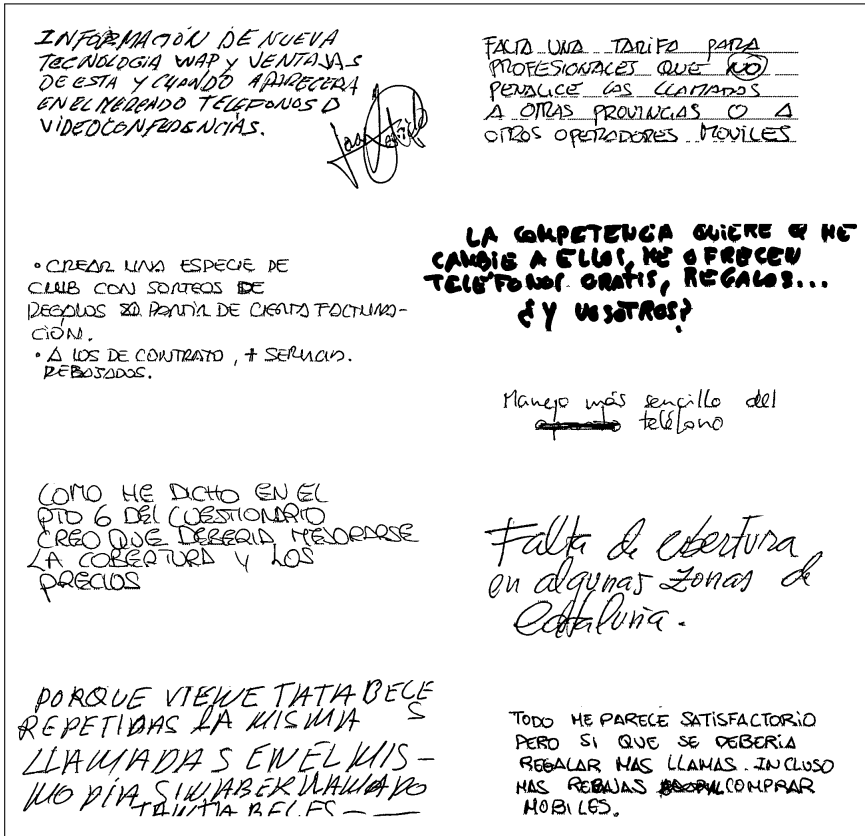


Figure 2.4: Examples of the answers to the suggestion point in the ODEC forms. In these examples we can see the differences on the stroke thickness, the irregular and non-consistent spacing between words and characters, the differences on the types and sizes of the words, words containing orthographic mistakes, crosses-out words, non-textual artefacts, etc.

Table 2.3: Basic statistics of the database ODEC.

Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
Writers/phrases	676	237	913	-	-	-
Words	12,287	4,084	16,371	2,790	518	4.4
Characters	64,666	21,533	86,199	80	0	808

2.4 IAM database

This corpus was compiled by the Research Group on Computer Vision and Artificial Intelligence (FKI) at Institute of Computer Science and Applied Mathematics (IAM) in Bern

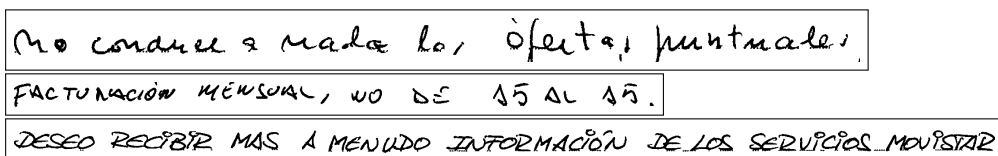


Figure 2.5: Three (short) sample sentences from the ODEC Database after adequately pasting the extracted lines.

(Switzerland). The database was first published in the ICDAR (International Conference of Document Analysis and Recognition) in 1999 [MB99]. In the second version of the database a segmentation scheme was used, which is documented in [ZB00]. The IAM-database as of October 2002 is described in [MB02]. It is publicly accessible and freely available upon request for non-commercial research purposes. The IAMDB images correspond to handwritten texts copied from the Lancaster-Oslo/Bergen (LOB) corpus [GLV95, JLG78], which encompasses around 500 printed electronic English texts of about 2,000 words each and about one million total running words.

The text in the LOB corpus is of quite diverse nature, because it is composed of different categories (editorial, reportage, religion, skills, hobbies, science fiction, ...). The LOB corpus (see [JLG78]) was compiled by researchers from Lancaster, Oslo and Bergen between 1970 and 1978. It contains about one million words in 500 printed texts of British English. The text in the corpus were split into fragments of about 3 to 6 sentences with at least 50 words each. These text fragments were copied onto forms and different persons were asked to write the text on one or more of the forms by hand. No restriction was imposed on the writing style or the type of pen to be used. The forms of unconstrained handwritten text were scanned at a resolution of 300dpi and saved as PNG images with 256 gray levels. On Figure 2.6 we can see an example of one of these forms.

The IAMDB version 3.0 (the latest at this moment) is composed of 1,539 scanned text pages, handwritten by 657 different writers. This dataset is provided at different levels: sentences, lines and isolated words. In this thesis we are going to work with the sentences partition [ZCB06]. Line detection and extraction, as well as (manually) detecting sentence boundaries, was carried out by the IAM institute [MB01]. Using this information, lines could be easily merged into whole sentence line-images. In Figure 2.7 we can see some examples of handwritten lines images from this corpus. All the information related with the partition used on this thesis is summarized on Table 2.4.

Table 2.4: Basic statistics of the database IAM.

Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
Writers	448	100	548	–	–	–
Sentences	2,124	200	2,324	–	–	–
Words	42,832	3,957	46,789	8,017	921	5.3
Characters	216,774	20,726	237,500	78	0	2,779

Sentence Database	A01-000
<hr/> <p>A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, M P for Manchester Exchange.</p> <hr/>	
<p>A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour MP's tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, MP for Manchester Exchange.</p>	
<hr/> <p>Name: _____</p>	

Figure 2.6: A sample form from the IAM Database.

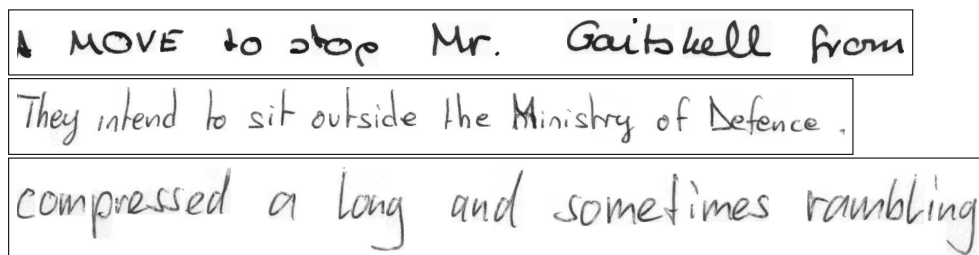


Figure 2.7: Some sample lines from the IAM Database.

Following [BB08], three additional text corpora have been used to train the n -grams and lexicon for the recognition task, namely, the full LOB corpus (except the sentences for the image test set), the Brown corpus and the Wellington corpus.

The Brown corpus was published by Francis and Kucera in 1964 [FK64]. It was the first modern computer readable corpus and it corresponds to the LOB corpus in size and content, but contains words in American English.

The third corpus, which also equals the LOB corpus in size and content is the Wellington corpus [HVJ98], it contains words in New Zealand English.

Table 2.5 gives more information of the three text corpora.

Table 2.5: Description of the text corpora.

Number of:	LOB	Brown	Wellington
Lines	52,676	49,362	56,745
Running words	1,119,904	1,045,213	1,144,401
Vocabulary size	52,724	53,115	58,919
Running Characters	5,803,916	5,582,023	6,055,820

2.5 UNIPEN

The UNIPEN Train-R01/V07 dataset is an English publicly available on-line HTR corpus^c. It comes organized into several categories [GSP+94] such as lower and upper-case letters, digits, symbols, isolated words and full sentences. According to the UNIPEN categorization, isolated digits category is identified as 1a, isolated lowercase letters category is identified as 1c and isolated symbols category is labelled with 1d.

This corpus is used here to test the multimodal text correction on the MM-CATTI system described in Chapter 5. Unfortunately, the UNIPEN isolated words category does not contain all (or almost none of) the required word instances to be handwritten by the user in the MM-CATTI interaction process with the ODEC, IAMDB, or CS text images. Therefore,

^cFor a detailed description of this dataset, see <http://www.unipen.org>.

the needed on-line handwritten words were generated by concatenating random character instances from three UNIPEN categories: 1a (digits), 1c (lowercase letters) and 1d (symbols). Some character examples from these categories are shown in Figure 2.8.

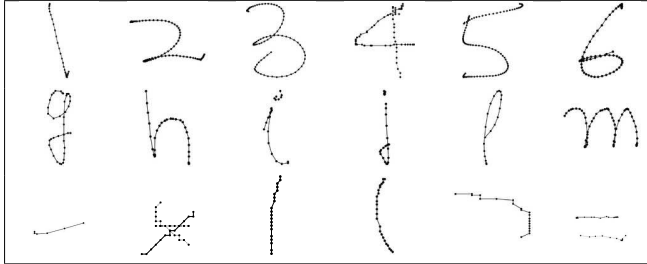


Figure 2.8: Some examples from the categories 1a,1c and 1d in the Train-R01/V07 dataset.

Table 2.6 shows the basic statistics of the different categories of the UNIPEN dataset and the corresponding partition definitions.

Table 2.6: Basic statistics of the UNIPEN categories 1a,1c and 1d in the Train-R01/V07 dataset and their corresponding partition definitions.

Number of:	Train	Test	Total	lexicon
digits (1a)	9,032	6,921	15,953	10
letters (1c)	39,354	18,894	58,248	26
symbols (1d)	10,321	6,849	17,170	32
All Together	58,707	32,664	91,371	68

Bibliography

- [BB08] R. Bertolami and H. Bunke. Hidden markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, 41(11):3452–3460, 2008.
- [Dri06] F. Drida. Towards restoring historic documents degraded over time. In *Proceedings of the Second International conference on Document Image Analysis for Libraries (DIAL'06)*, IEEE Computer Society, pages 350–357. Washington, DC, USA, 2006.
- [FK64] W.N. Francis and H. Kucera. *Manual of Information to Accompany a Standard Corpus of Present-Day Edited American English, for use with Digital Computers*. Department of Linguistics, Brown University, Providence, Rhode Island, USA, 1964.
- [GLV95] R. Garsid, G. Leech, and T. Váradi. *Manual of Information for the Lancaster Parsed Corpus*. Bergen, Norway: Norwegian Computing Center for the Humanities, 1995.
- [GSP+94] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks. In *Proceedings of the 14th International Conference on Pattern Recognition*, pages 29–33, Jerusalem (Israel), 1994.
- [HVJ98] J. Holmes, B. Vine, and G. Johnson. *Guide to the Wellington Corpus of Spoken New Zealand English*. School of Linguistics and Applied Language Studies, Victoria University of Wellington, Wellington, New Zealand, 1998.
- [JLG78] S. Johansson, G. Leech, and H. Goodluck. *Manual of Information to Accompany the Lancaster-Oslo/bergen Corpus of British English, for Use with Digital Computers*. Dept. of Englis, Univ. of Oslo, Norway, 1978.
- [MB99] U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *Proceedings of the International Conference Document Analysis and Recognition (ICDAR'99)*, pages 705–708, Bangalore (India), 1999.
- [MB01] U.-V. Marti and H. Bunke. Using a Statistical Language Model to improve the performance of an HMM-Based Cursive Handwriting Recognition System. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.
- [MB02] 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*, 5:39–46, 2002.
- [TJV04] Alejandro H. Toselli, Alfons Juan, and Enrique Vidal. Spontaneous Handwriting Recognition and Classification. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 433–436, Cambridge, United Kingdom, August 2004.

- [Tos04] Alejandro Héctor Toselli. *Reconocimiento de Texto Manuscrito Continuo*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia (Spain), March 2004. Advisor(s): Dr. E. Vidal and Dr. A. Juan (in Spanish).
- [ZB00] M. Zimmermann and H. Bunke. Automatic segmentation of the iam off-line database for handwritten english text. In *In Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 35–39, 2000.
- [ZCB06] M. Zimmermann, J.-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):818–821, 2006. Member-Horst Bunke.

CHAPTER 3

Handwritten Text Recognition

3.1 Introduction

Handwritten text recognition (HTR) is not an easy task. The difficulties to segment text lines, the variability of the handwriting, the complexity of the styles and the open vocabulary make that the HTR systems do not obtain acceptable results on unconstrained handwritten documents. However, humans being seem to carry out this process effortlessly and in a natural way. Observing the way in which the humans do this recognition, it seems clear that this human ability is due to the inter-cooperation between different levels of knowledge. These levels are: morphological, lexical and syntactical knowledge.

The situation in automatic speech recognition (ASR) is very similar [Jel98]. So, it is natural that both ASR and HTR are tackled with techniques based on the cooperation of all the aforementioned knowledge sources [BS89, EYGSS99, SR98, PS00, MB01]. The different knowledge levels can be modelled using finite state models, such as HMMs, grammars or automata.

As mentioned in Chapter 1, depending on the mode of data acquisition used, automatic handwriting recognition can be classified into on-line and off-line. In this thesis the main system (see Chapter 4) is an *off-line* system. However, the feedback provided by the user in the MM-CATTI (see Chapter 5) comes in the form of *on-line* data. In this chapter a general overview of both off-line and on-line HTR systems is presented in Sections 3.2 and 3.3, respectively. The off-line system is applied in three tasks of different nature. These tasks involve the transcription of handwritten answers from survey forms, handwritten full English sentences of different categories (editorial, religion, fiction, love, humour, ...) and a legacy handwritten document written in the year 1853. On the other hand, the on-line system is

tested on an English dataset divided into several categories: letters, digits, symbols and isolated words (see Chapter 2).

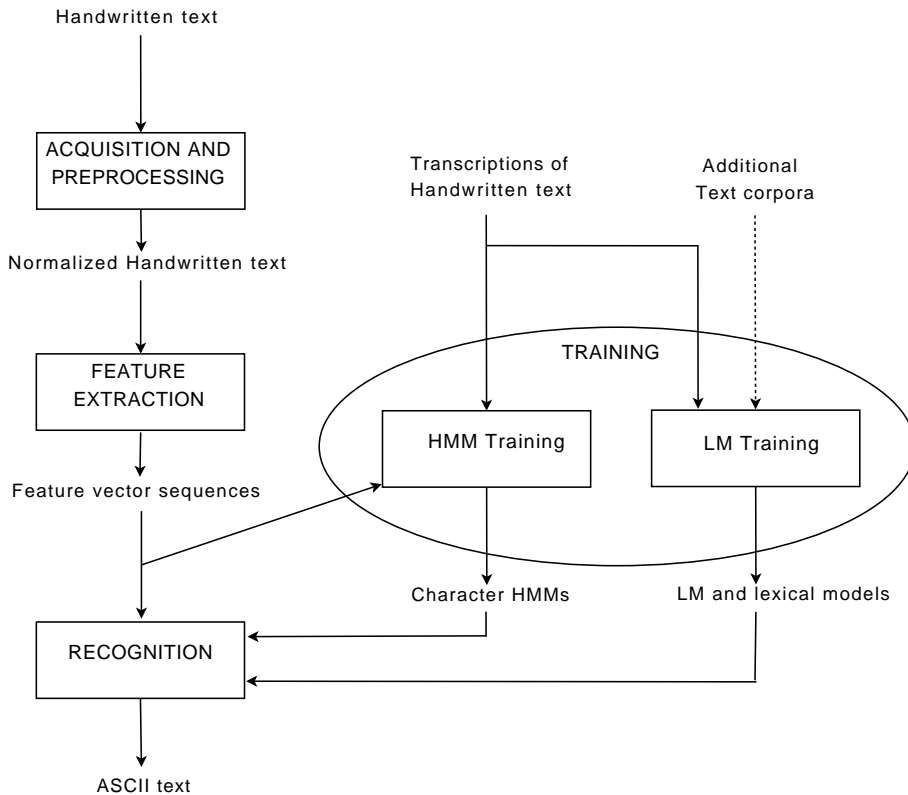


Figure 3.1: Overview of the HTR system.

Both, the off-line and the on-line HTR systems presented in this chapter follow the classical architecture composed of three main modules: preprocessing, feature extraction and recognition. The preprocessing module is in charge to reduce the variability of text styles. In the feature extraction module a feature vector sequence is obtained as the representation of a handwritten text image or pen trajectory. And the recognition module obtains the most likely word sequence for the sequence of feature vectors. In addition, there is another module in charge to train the different models used on the recognition step: Hidden Markov models, language models and lexical models. On Figure 3.1 we can see a diagram of this architecture. It needs to be mentioned that the recognition module used in this chapter will be adapted in Chapter 4 in order to cope with interactively produced validated prefixes.

3.2 Off-line Handwritten Text Recognition

This section presents the off-line HTR system used in this work. The following three subsections describe each of the main modules (preprocessing, feature extraction and recognition) in detail. Then, the different experiments carried out, the assessment measures used and the results obtained are explained in Subsection 3.2.4 and 3.2.5. Finally, some conclusions are drawn in Section 3.2.6.

3.2.1 Preprocessing

It is quite common for handwritten documents, and above all for ancient documents, to suffer from degradation problems [DRI06]. Among these are the presence of smear, background of big variations and uneven illumination, spots due to the humidity or marks resulting from the ink that goes through the paper (often called bleed-through). In addition, other kinds of difficulties appear in these pages as different font types and sizes in the words, underlined and/or crossed-out words, etc. The combination of these problems contributes to make the recognition process difficult, therefore a preprocessing module becomes essential.

There is not a general, standard solution to carry out the preprocessing, and it can be said that each handwriting recognition system has its own, particular solution. In the preprocessing module used in this work the following steps take place: background removal and noise reduction, skew correction, line extraction, slope correction, slant correction and size normalization.

Note that IAMDB and ODEC corpora are provided at sentence level, so the skew correction and line extraction operations will only be applied at the CS corpora.

Background removal and noise reduction

It is quite common in handwritten documents to suffer from degradation problems. These problems are factors that impede (in many cases may disable) the legibility of the documents. Therefore, appropriate filtering methods should be developed in order to remove noise and improve their quality and do the documents more legible. Within this framework, noise is considered anything that is irrelevant for the textual information (i.e, the foreground) of the document image.

In this work, background removal and noise reduction are performed by applying a 2-dimensional median filter [KS06] on the entire image and subtracting the result from the original image. Then, a grey-level normalization to increase the foreground/background image contrast is applied (see Figure 3.2).

Skew correction

The skew is a distortion introduced during the document scanning process. It is understood as the angle of the document paper with respect to the scanner coordinates system. So, skew correction must be carried out on each whole document image, by aligning their text lines with the horizontal direction. This process makes it easy the process of line and paragraph extraction. The skew correction is carried out by searching for the angle which maximizes

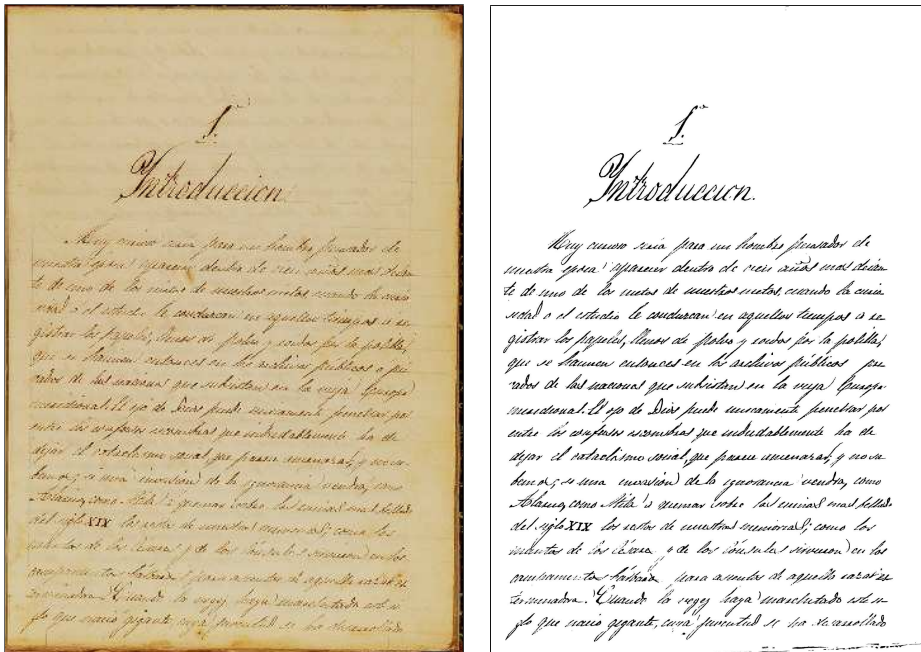


Figure 3.2: Background removal, noise reduction and skew correction example: a) original image; b) skew correction, background removal, noise reductions and increase of contrast.

the variance of the horizontal projection profile and then applying a rotation operation with this angle [iG07].

Some authors often refer to this misframing as skew [AF00, YCL03], other authors define skew as the angle between the horizontal direction and the direction of the line on which the writer aligns the words, what is called slope in this work [MB01, Vin02, MMS99], and some authors use it interchangeably to refer to the misframing of the entire page, and for the wrong alignment of the words with the horizontal direction.

Figure 3.2 (right) illustrates a page image after background removal, noise reduction, increase of contrast and skew correction.

Line extraction

The next step consists in dividing the page image into separate line images. The method used is based on the horizontal projection profile of the input image. Local minima in this projection are considered as potential cut-points located between consecutive text lines. When the minimum values are greater than zero, no clear separation is possible. This problem has been solved using a method based on connected components [MB01].

Figure 3.3 (right) shows the resulting line images extracted from the highlighted region

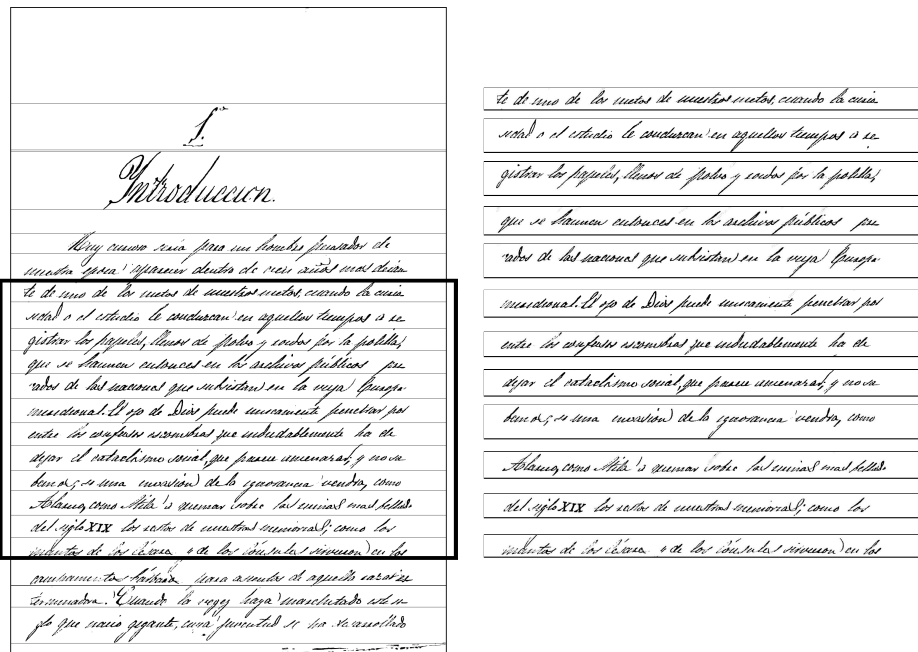


Figure 3.3: Line extraction example: Left) Image with cutting lines; Right) separated line images from the highlighted region.

in the page image of the left after applying the method above mentioned. In Figure 3.4 (top) we can see in more detail one of the resulting line images.

Slope correction

The slope is the angle between the direction of the line on which the writer aligned the words on a text line and the horizontal direction. The slope correction processes an original image to put the text line into horizontal position by applying a rotation operation with the same slope angle, but in the opposite direction. As each word or multiword segment in the text line may have a different slope angle, the original image is divided into segments surrounded by wide blank spaces and slope correction is applied to each segment separately. This is not to obtain a segmentation of the text line into words as it is not necessary for each segment to contain exactly one word.

This division is usually based on more or less sophisticated heuristics. A minimum size of blank space is used to define segments of words. In this work, to define this minimum blank space a vertical projection of the image is carried out, and the average size of all the spaces is computed. All space of the projection for which the gray level is below a threshold will be considered blank space.

To obtain the angle we use a method based on horizontal projections, very similar to the method used on the skew correction operation [iG07, TJK+04].

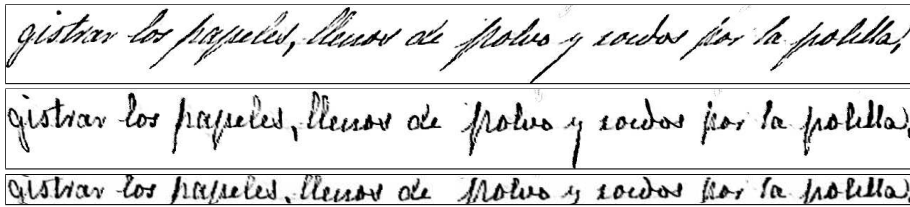


Figure 3.4: Slant and size normalization example: Top) A separated line image; Middle) Slant correction; Bottom) Size normalization.

Slant correction

Slant correction shears the isolated line image horizontally to bring the writing in an upright position. The slant angle is determined using a method based on vertical projection profile (see [PTV04, PTRV06]). The method is based on the observation that the columns distribution of the vertical projection profile presents a maximum variance for the non slanted text. Figure 3.4 (middle) shows an example of an image after applying the slant correction operation.

Size normalization

The purpose of the this preprocessing operation is to make the system invariant to character size and to reduce the areas of background pixels which remain on the image because of the ascenders and descenders of some letters. Since each word (or group of words) of the main body can have a different height, the original image is divided into segments surrounded by wide blank spaces in the same way that it is explained on the “slope correction”. Then, upper and lower lines of each segment are computed. For this purpose, the algorithm “Run-Length Smoothing Algorithm” (RLSA) [KYWW82] is applied and upper and lower contours for each segment are detected. Then, eigenvector line fitting algorithm [DH73] is applied to the contour points to compute upper and lower baselines. These lines separate the whole main text body from the zones including ascenders and descenders.

In order to obtain an uniform text line, i.e. where all their segments have the same height, it is necessary to set the same normalization height for all the main body segments. To this end, we compute the average of the size of the main body of all the segments and linearly scale the main body heights of each segment to this value. Finally, the ascenders and descenders are linearly scaled in height to a size determined as an empirically chosen percentage of the new main body vertical size (30% for ascenders and 15% for descenders). Since these zones are often huge blank areas, this scaling operation has the effect of filtering out most of the uninformative image background. It also accounts for the large height variability of the ascenders and descenders as compared with that of the main text body [RPTV06].

The image on the bottom of the Figure 3.4 shows this last step in the preprocessing module.

3.2.2 Feature extraction

Our HTR system is based on Hidden Markov Models (HMMs), so each preprocessed text line image has to be represented as a sequence of feature vectors. Several approaches have been proposed to obtain this kind of sequences [BSM99, MB01, BRKR00]. The approach used in this work follows the ideas described in [BSM99].

First, given a text line image of $n_r \times n_c$ pixels, a grid is applied to divide the text line image into $N \times M$ squared cells. N is chosen empirically and $M = \rho \frac{n_c N}{n_r}$, i.e. M is chosen so that M/N is proportional to the original normalized line image aspect ratio [TJK+04]. ρ is the factor of proportionality and its value is chosen empirically. Each cell is characterized by the following features:

1. Normalized gray level
2. Horizontal gray level derivative
3. Vertical gray level derivative

The normalized gray level provides information about the trace density on the analysed cell. On the other hand, the derivatives give information about how this density varies along different directions.

To obtain smoothed values of these features, feature extraction is not restricted to the cell under analysis, but extended to a $r \times s$ cells window, centered at the current cell. The values of r and s are chosen empirically.

To compute the normalized gray level, the analysis window is smoothed by convolution with a 2-d Gaussian filter. So, the gray level of each pixel affected by the Gaussian, on an analysis window with $n \times m$ pixels, is computed as:

$$I'(x, y) = I(x, y) \exp \left[-\frac{1}{2} \left(\frac{x - (m/2)^2}{(m/4)^2} + \frac{y - (n/2)^2}{(n/4)^2} \right) \right]$$

Finally, the gray level for the cell under analysis is computed as the average of the gray level from the pixels on the window:

$$g = \frac{\sum_{i=1}^m \sum_{j=1}^n I'(x_i, y_j)}{nm}$$

The horizontal derivative is calculated as the slope of the line which best fits the horizontal function of column-average gray level. The fitting criterion is the sum of squared errors weighted by a 1-d Gaussian filter which enhances the role of central pixels of the window under analysis. The vertical derivative is computed in a similar way. The average gray level for each column is computed as:

$$g(x_i) = \frac{\sum_{j=1}^n I(x_i, y_j)}{n}$$

Linear regression is applied in order to determine the straight line $ax = b$ which distance J to the points $(x_i, g(x_i))$ is minimum:

$$J = \sum_{i=1}^n w_i (g(x_i) - (ax_i + b))^2$$

where w_i is a Gaussian filter function for the x_i position:

$$w_i = \exp\left(-\frac{1}{2} \frac{(x_i - m/2)^2}{(m/4)^2}\right)$$

the computed value of the parameter a (straight line slope) is the horizontal derivative value for the cell under analysis:

$$a = \frac{(\sum_{i=1}^m w_i g(x_i))(\sum_{i=1}^m w_i x_i) - (\sum_{i=1}^m w_i)(\sum_{i=1}^m w_i g(x_i) x_i)}{(\sum_{i=1}^m w_i x_i)^2 - (\sum_{i=1}^m w_i)(\sum_{i=1}^m w_i x_i^2)}$$

Columns of cells (also called *frames*) are processed from left to right and a feature vector is constructed for each *frame* by stacking the three features computed in its constituent cells. Hence, at the end of this process, a sequence of M $\{N \times 3\}$ -dimensional feature vectors (N normalized gray-level components and N horizontal and vertical derivatives components) is obtained. This process is similar to that followed in [BSM99]. In Figure 3.5 an example of the feature vectors sequence for a fraction of a separate line image is shown graphically.



Figure 3.5: Example of the feature vectors sequence for a portion of a separated line image.

3.2.3 Recognition

Probabilistic Framework

As explained before, a handwritten sentence image can be represented by a sequence of any number M of feature vectors of dimension $N \times 3$, $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_M)$ $x_i \in \mathfrak{R}^{N \times 3}$. Therefore, the traditional handwritten text recognition problem can be formulated as the problem of finding the most likely word sequence, $\mathbf{w} = (w_1 \ w_2 \ \dots \ w_l)$, for the given handwritten sentence image represented by the feature vector sequence \mathbf{x} :

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{w}|\mathbf{x}) \tag{3.1}$$

Using the Bayes' rule:

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} \frac{\Pr(\mathbf{x}|\mathbf{w}) \Pr(\mathbf{w})}{\Pr(\mathbf{x})} \quad (3.2)$$

Since, $\Pr(\mathbf{x})$ is the same for any likely word sequence \mathbf{w} , the probability $\Pr(\mathbf{w}|\mathbf{x})$ is decomposed into two probabilities as shown in equation (3.2), $\Pr(\mathbf{x}|\mathbf{w})$ and $\Pr(\mathbf{w})$, representing morphological-lexical knowledge and syntactic knowledge, respectively:

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} \Pr(\mathbf{x}|\mathbf{w}) \Pr(\mathbf{w}) \approx \operatorname{argmax}_{\mathbf{w}} P(\mathbf{x}|\mathbf{w})P(\mathbf{w}) \quad (3.3)$$

$\Pr(\mathbf{x}|\mathbf{w})$ is typically approximated by concatenated character models, usually Hidden Markov Models [Jel98, Rab89], and $\Pr(\mathbf{w})$ is approximated by a word language model, usually n -grams [Jel98].

In practice, the simple multiplication of $P(\mathbf{x}|\mathbf{w})$ and $P(\mathbf{w})$ needs to be modified in order to balance the absolute values of both probabilities. The most common modification is to use a language weight α (Grammar Scale Factor, GSF), which weights the influence of the language model on the recognition result, and an insertion penalty β (Word Insertion Penalty, WIP), which helps to control the word insertion rate of the recognizer [OTI98]. In addition, we are going to use log-probabilities to avoid the numeric underflow problems that can appear using probabilities. So, the problem consists on finding the word sequence $\hat{\mathbf{w}}$ that maximizes the score $\varphi(\mathbf{w})$:

$$\varphi(\mathbf{w}) = \log P(\mathbf{x}|\mathbf{w}) + \alpha \log P(\mathbf{w}) + l\beta \quad (3.4)$$

where l is the word length of the sequence \mathbf{w} . Now, Equation (3.3) can be rewritten as:

$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} \log P(\mathbf{x}|\mathbf{w}) + \alpha \log P(\mathbf{w}) + l\beta \quad (3.5)$$

where α and β are optimized for all the training sentences of the corpus.

Character, Word and Language Modelling

HMMs have received significant attention in handwriting recognition during the last few years. As in speech recognizers for acoustic data [aBHJ93], HMMs are used to estimate the probability for a sequence of feature vectors representing a handwritten text image. Sentences models are built by concatenation of word models which, in turn, are often obtained by concatenation of continuous HMMs for individual characters.

Basically, each character HMM is a stochastic finite-state device that models the succession, along the horizontal axis, of feature vectors which are extracted from instances of this character (see Section 1.4.1 for a formal definition of HMM). Each HMM state generates feature vectors following an adequate parametric probabilistic law; typically, a Gaussian Mixture.

The required number of Gaussians in the mixture depends, along with many other factors, on the vertical variability typically associated with each state. On the other hand, the adequate number of states to model a certain character depends on the underlying horizontal variability. For instance, to ideally model a capital "H" character, only three states might be

enough (one to model the first vertical bar, other for the horizontal line, and finally the other for the last vertical bar). Note that the possible or optional blank space that may appear between characters should be also modelled by each character HMM. The number of states and Gaussians define the total amount of parameters to be estimated. Therefore, these numbers need to be empirically tuned to optimize the overall performance for the given amount of training vectors available.

Taking this into account, the HMMs have two implicit stochastic process: one of them describes the different parts that conform the object (character) and the other the variability of these parts. For HTR it is adequate to use left-to-right HMMs, appropriate to the horizontal direction of extraction process of feature vector sequence. In this models a transition between two states $q_i, q_j \in Q$ from the HMM, it is only possible if $j \geq i$. Figure 3.6 shows an example of how a HMM models two feature vector subsequences corresponding to the character “a”.

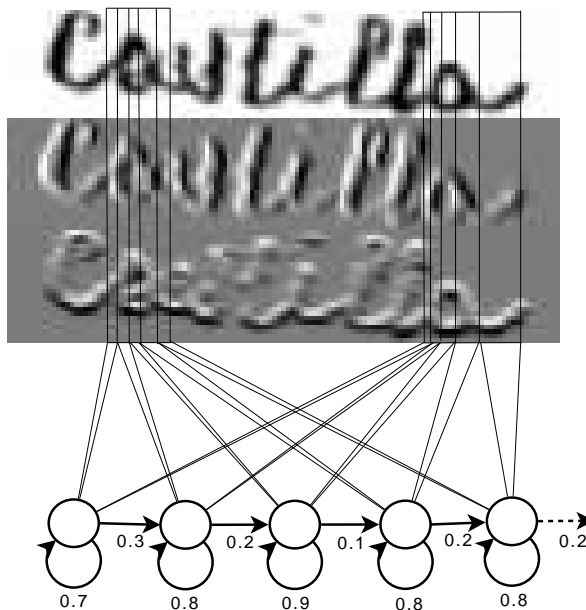


Figure 3.6: Example of 5-states HMM modelling (feature vectors sequences of) instances of the character “a” within the Spanish word “Castilla”. The states are shared among all instances of characters of the same class.

Once an HMM topology (number of states and structure) has been adopted, the model parameters can be easily trained from images of continuously handwritten text (without any kind of segmentation) accompanied by the transcription of these images into the corresponding sequence of characters. This training process is carried out using a well known instance of the EM algorithm called forward backward or Baum-Welch (see Section 1.4.1).

The concatenation of characters to form words is modelled by simple lexical models. Each word is modelled by a stochastic finite-state automaton which represents all possible

concatenations of individual characters that may compose the word. This automaton takes into account optional character capitalizations. By embedding the character HMMs into the edges of this automaton, a *lexical HMM* is obtained. These HMMs estimate the word-conditional probability $P(\mathbf{x}|\mathbf{w})$ of Equation (3.3). An example of automaton for the word “the” is shown in Figure 3.7

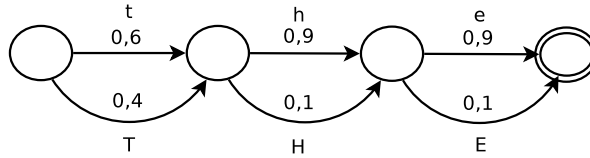


Figure 3.7: Automaton for the lexicon entry “the”.

Finally, the concatenation of words into text lines or sentences is modelled by an n -gram *language model*, with Kneser-Ney back-off smoothing [Kat87, KN95], which uses the previous $n - 1$ words to predict the next one; that is,

$$\Pr(\mathbf{w}) \approx \prod_{i=1}^N P(w_i | \mathbf{w}_{i-n+1}^{i-1}) \quad (3.6)$$

N -gram models estimate the probability $\Pr(\mathbf{w})$ in Equation (3.3). A simple example of LM is shown in Figure 3.8

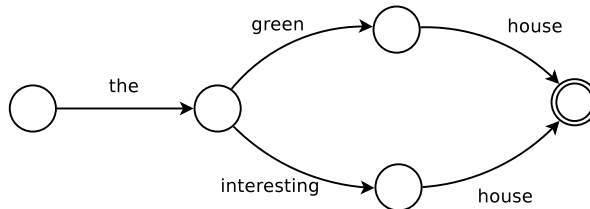


Figure 3.8: A simple language model.

Once all the *character*, *word* and *language* models are available, recognition of new test sentences can be performed. Thanks to the homogeneous finite-state (FS) nature of all these models, they can be easily *integrated* into a single *global* FS model by replacing each word character of the n -gram model by the corresponding HMM. The search for decoding the input feature vectors sequence \mathbf{x} into the output words sequence $\hat{\mathbf{w}}$, is performed over this global model. This search is optimally done by using the Viterbi algorithm [Jel98]. This algorithm can be easily adapted also for the search required in the CATTI interactive framework explained in Chapter 4.

All the above mentioned things have been made at word level. However, it can be similar done at character level by replacing \mathbf{w} with \mathbf{c} at the Equation (3.4). So, now the problem can

be formulated as the problem of finding the most likely character sequence \mathbf{c} , for the given feature vector sequence \mathbf{x} :

$$\hat{\mathbf{c}} = \operatorname{argmax}_{\mathbf{c}} \Pr(\mathbf{x}|\mathbf{c}) \Pr(\mathbf{c}) \approx \operatorname{argmax}_{\mathbf{c}} P(\mathbf{x}|\mathbf{c})P(\mathbf{c}) \quad (3.7)$$

$\Pr(\mathbf{x}|\mathbf{c})$ can be modelled by Hidden Markov Models and $\Pr(\mathbf{c})$ is approximated by n -grams. In this case, the language model will model the way in which the characters can be concatenated into text lines or sentences. On the other hand, the lexicon model will be describing the different characters that can be recognized during the recognition process.

3.2.4 Experimental Framework

The experimental framework adopted to assess the effectiveness of the HTR system presented here is described in the following subsections. These subsections include some details of the different corpora, the performance measures used and a description of the values used for the different parameters tested on the experiments.

Corpora

Experiments have been carried out using the three different off-line corpora explained in the Chapter 2.

To train the character HMMs of the different corpora, we consider all the elements appearing in each handwritten text image of the training set, such as lowercase and uppercase letters, symbols, abbreviations, spacing between words and characters, crossed-words, etc. However, to train the n -gram models we imposed certain usual/useful restrictions intended to simplify the experiments and additionally allowing them to be fully reproducible. This restrictions consist in converting uppercase character to lowercase and eliminating the punctuation signs. On CS and ODEC corpora only the text corresponding to the training partition has been used to train the n -grams. However, the extra words that appear on the test partition are added to the 1-gram. In this way, we are working with closed vocabulary. The number of words of the test partition that do not appear on the train partition is represented in the column OOV of the tables that summarize the corpora data in the Chapter 2.

For the IAM Dataset the training of the n -grams is not restricted only to the text transcription of the train sentences, but the full LOB, Brown and Wellington text corpora have been used after removing the test sentences.

Assessment Measures

Two different measures have been used to asses the quality of the transcriptions obtained with the HTR system. These measures are the well known word error rate (WER) and the character error rate (CER).

The WER counts the number of different words between the transcription proposed by the system and the reference transcription. This measure has been used to assess the accuracy of automatic speech recognition (ASR) systems. The computation of this measure is not trivial, because the hypotheses length can be different from the reference length. Therefore, the WER

is defined by means of an alignment between the two word sequences. In this alignment 4 different situations can occur:

- **correct word:** the reference word coincides with the aligned hypothesis word.
- **substitution:** the reference word is aligned with a different word from the hypothesis.
- **insertion:** a word has been inserted in the hypothesis that can not be aligned with any word from the reference.
- **deletion:** a reference word does not appear in the hypothesis.

The optimal alignment is defined as the alignment that minimizes the *Levenshtein* distance [SK83]; i.e., the minimum number of insertions, deletions and substitutions between the two word sequences. This value can be obtained using dynamic programming. WER can be defined as the minimum number of words that need to be substituted, deleted or inserted to convert a sentence recognized by the system into the corresponding reference transcription, divided by the total number of words in the reference transcription:

$$WER = 100 \cdot \frac{N_i + N_s + N_d}{N_s + N_d + N_c}$$

where N_i is the number of insertions; N_s is the number of substitutions; N_d is the number of deletions and N_c is the number of correct words.

Note, that if the number of insertion is too high, the WER can be greater than 100%.

The definition of the CER is analogous to the previous one, but substituting words by characters.

Parameters

There are some basic parameters that need to be adjusted to design an accurate recognizer with our approach. They are:

- N is the vertical number of cells in which the image is divided.
- ρ is the factor of proportionality to the original normalized line image aspect ratio. This value is used to compute the number of horizontal cells in which the image is divided (M) computed as: $M = \rho \frac{n_c N}{n_r}$, i.e. M .
- $r \times s$ is the size of the analysis window used during the feature extraction process.
- N_S is the number of states for each character HMM.
- N_G is the number of Gaussian densities used in each state of the HMM.
- α is the grammar scale factor (GSF)
- β is the word insertion penalty (WIP)

Automatic determination of optimal values for these parameters is not an easy task. In particular, it is difficult to determine independent, optimal values of N_S and N_G for each character HMM. For simplicity, we decided to use the same values of N_S and N_G for all HMMs. The best parameter values were empirically tuned for each task.

Feature extraction was applied to the preprocessed databases to obtain a sequence of M ($3N$)-dimensional feature vectors for each handwritten image. As discussed in Section 3.2.3, left to right continuous density HMMs of N_S states and N_G Gaussian densities per state were used for character modelling. Each HMM has only two transitions per state (one at the same state and the other one to the next state). In addition, only diagonal covariance matrix for each Gaussian on the mixture were used. These HMMs were trained through four iterations of the Baum-Welch algorithm. This algorithm was initialized by a linear segmentation of each training image into a number of equal-length segments (according to the number of characters in the orthographic transcription of the sentence). For each test sentence, the Viterbi algorithm was performed on the integrated finite-state network to obtain the desired recognized transcription.

Another parameter that must be adjusted is the order of the n -gram language model. Here, we have carried out experiments using language models at word level and at character level. Taking into account previous results we decided to test the word language model using bi-grams. However, to test the language model at character level orders ranging from 2 to 9 were used.

3.2.5 Results

Results with word language models

We have carried out experiments with the different off-line corpora presented at Chapter 2. First we can see the results obtained with the corpora Cristo-Salvador.

Figure 3.9 shows the HTR WER(%) obtained for the corpora Cristo Salvador in its page partition as a function of the number of states per HMM ($N_S = 6, 8, 10, 12, 14$) and for different values of the parameter $N = 16, 20, 24$, fixing $\rho = 2$ (left) and $\rho = 3$ (right). The size of the analysis windows used here is 9×5 , that is the size that has provided the best result. On the appendix B the complete set of experiments carried out is shown.

The best WER, 30.5%, has been obtained using $N = 20$, $N_S = 12$ and $\rho = 3$. From now on, we are going to use this baseline to continue looking for the best value of the parameters N_G , α and β . The tested values of this parameters are: $N_G = 32, 64, 128$; $\alpha = 60, 70, 80, 90$ and $\beta = -140, -160, -180$. On Figure 3.10 we can see the evolution of WER for an increasing value of the parameter α and different values of the parameter β , fixing N_G to 32. The results for $N_G = 64$ and 128 are shown on Appendix B. The best result, 28.5% is obtained for $\alpha = 90$ and $\beta = -140$. This result will be the baseline of the experiments carried out on the next chapters.

For the CS corpora on its book partition we assume that the best values of the parameters for the feature extraction module are the same previously tuned on the CS corpora for its page partition. It is because we are using the same corpora. However, the bi-gram language model trained in this case will be different to the bi-gram trained on the page partition. So, we need to tune the parameters α and β for this partition. On Figure 3.11 we can see the

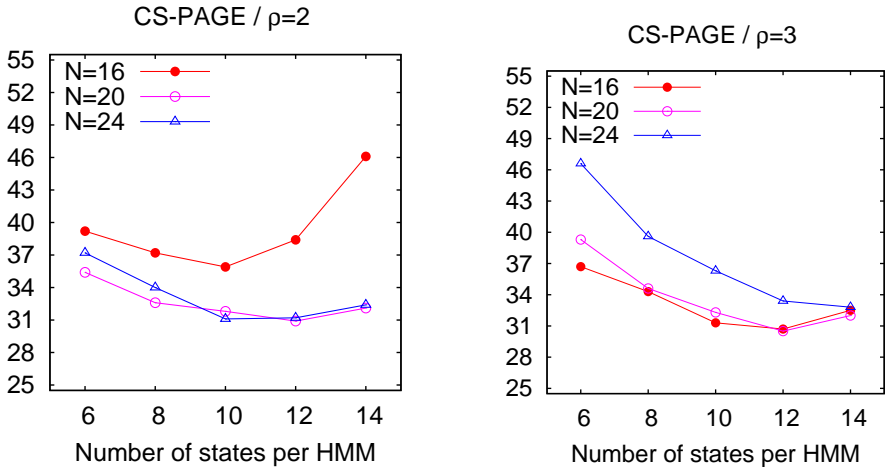


Figure 3.9: WER (%) for varying number of states per HMM and different values of the parameter N . $\rho = 2$ has been used on the left graph, and $\rho = 3$ on the right. In both cases, $r \times s = 9 \times 5$ has been used.

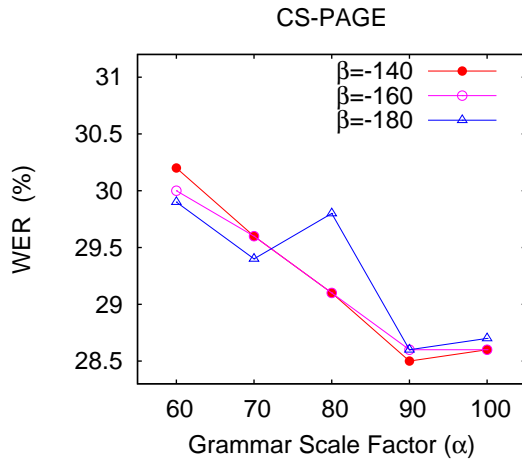


Figure 3.10: WER for different values of the parameter α and different values of the parameters β on the page partition of the CS corpora.

WER obtained as a function of the parameter α and for different values of the parameter β . The best result, 33.5% is obtained for the values $\alpha = 80$ and $\beta = -160$. This value will be used as baseline for the experiments carried out on the next chapters. Additional experiments can be seen on Appendix B.

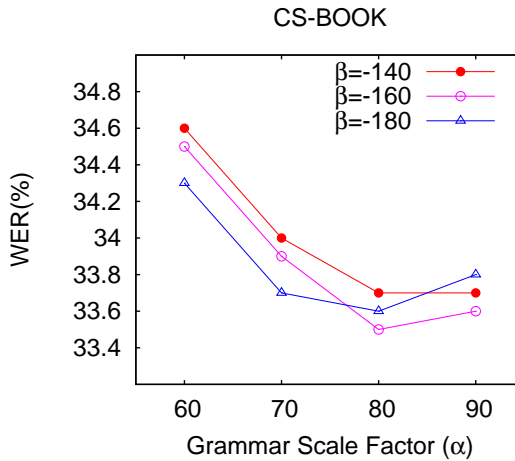


Figure 3.11: WER for different values of the parameter α and different values of the parameters β on the book partition of the CS corpora.

The results obtained with the CS corpora in its page partition are better than the results obtained in the book partition. It is mainly due to the fact that the test text line samples of the page partition were extracted from the same pages that the training text line samples. However, book partition better approaches a realistic transcription process as was explained in Section 2.2.

Similar experiments have been carried out with the other two corpora, where the tested values of the parameters of the feature extraction module have been the same for both ODEC and IAMDB corpora. The tested values for the parameter N are: 16, 20 and 24, and $\rho = 1$ and 2. The different tested values for the analysis windows are: 3×3 , 3×5 , 5×3 , 5×5 , 5×9 , 9×5 and 9×9 . Finally the number of states are : $N_S = 4, 6, 8$. Tables 3.1 and 3.2 show the HTR WER(%) obtained for the ODEC and IAMDB corpora respectively for the different values of the parameters N_S and N , being $\rho = 1$, $r \times s = 5 \times 3$ on the ODEC corpora and 5×5 on the IAMDB corpora. The rest of the experiments are shown on Appendix B.

Table 3.1: WER of the basic off-line HTR system for different values of the parameters N_S and N , fixing $\rho = 1$ and $r \times s = 5 \times 3$ on the ODEC corpora. All results are percentages.

N_S	N		
	16	20	24
4	27.5	33.7	45.6
6	23.2	24.1	29.3
8	41.4	26.6	25.3

The values of the parameters that obtain the best result on Tables 3.1 and 3.2 are fixed on

Table 3.2: WER of the basic off-line HTR system for different values of the parameters N_S and N , fixing $\rho = 1$ and $r \times s = 5 \times 5$ on the IAMDB corpora. All results are percentages.

N_S	N		
	16	20	24
4	35.3	34.2	40.6
6	29.4	25.3	26.5
8	41.0	25.9	25.8

the experiments to look for the best N_G , α and β . On Figures 3.12 and 3.13 we can see the obtained WER for ODEC and IAMDB as a function of the Grammar Scale Factor, α , and for different values of the Word Insertion Penalty, β . The values of the parameters used on the ODEC corpora to obtain the best result (22.9%) are: $N = 16$, $\rho = 1$, $r \times s = 5 \times 3$, $N_S = 6$, $N_G = 32$, $\alpha = 20$ and $\beta = -10$. For the IAMDB, the best result (25.3%) is obtained using $N = 20$, $\rho = 1$, $r \times s = 5 \times 5$, $N_S = 6$, $N_G = 64$, $\alpha = 40$ and $\beta = 0$.

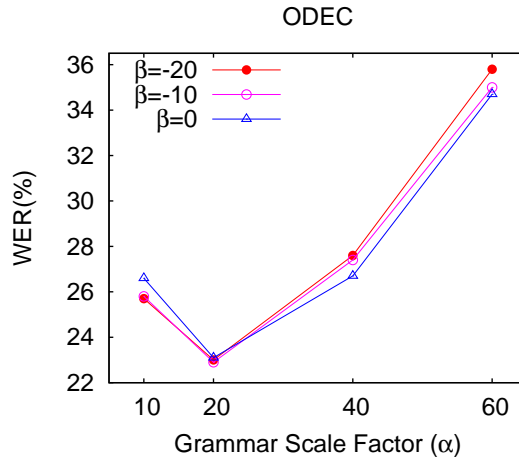


Figure 3.12: WER for different values of the parameter α and different values of the parameters β on the ODEC corpora.

Note that the best WER obtained for IAMDB (25.3%) is comparable with state-of-the-art results published for this data-set. Specifically in [ZCB06], similar results are reported using a system based also on HMMs and n -grams, but with a completely different feature extraction approach.

Table 3.3 summarized the best results obtained with the different corpora studied in this section.

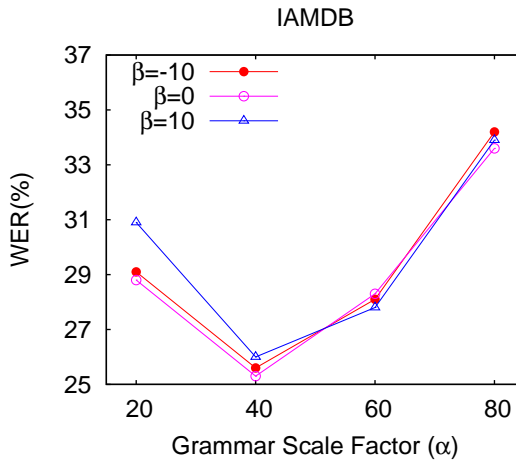


Figure 3.13: WER for different values of the parameter α and different values of the parameters β on IAMDB corpora.

Table 3.3: WER of the basic off-line HTR system for different corpora. All results are percentages.

CS-page	CS-book	ODEC	IAMDB
28.5	33.5	22.9	25.3

Results with character language models

To carry out the experiments with character language models we have used the HMMs that best results have obtained in the previous section. Then, the parameters α and β have been tuned for each n -gram order. Table 3.4 shows the best results (at character level CER) obtained for each n -gram order after tuning the α and β . The column WLM (Word Language Model) is the CER obtained with the best parameters values using a language model at word level, that correspond with the best values obtained previously.

Table 3.5 is analogous to the Table 3.4. However, in this case the shown results are at word level, i.e the best WER% obtained using character language models for each n -gram order after tuning the α and β is shown.

We can see on the results that using word language models work better than using language models at character level. It is due to the fact that the word language model provide better information about how characters and words must be concatenated to obtain correct sentences.

Table 3.4: CER obtained with different n -grams at character level. All results are percentages.

Corpora	n -gram order								WLM
	2	3	4	5	6	7	8	9	
ODEC	31.7	26.6	22.6	21.4	21.4	21.7	21.8	22.1	12.8
IAM-sentences	37.4	32.0	27.6	25.2	24.3	24.5	24.6	24.6	14.8
CS-page	33.1	29.6	27.2	25.7	25.7	25.5	25.6	25.6	15.4
CS-book	29.7	26.5	24.3	23.7	23.8	23.8	23.9	23.9	18.2

Table 3.5: WER obtained with different n -grams at character level. All results are percentages.

Corpora	n -gram order								WLM
	2	3	4	5	6	7	8	9	
ODEC	64.4	60.9	49.8	46.7	46.7	46.6	46.9	47.6	22.9
IAM-sentences	85.6	68.6	55.6	47.8	46.2	46.3	46.9	46.9	25.3
CS-page	72.8	63.0	51.1	50.8	51.0	50.9	51.0	51.1	28.5
CS-book	67.7	59.3	51.7	49.6	49.8	49.6	49.9	50.2	33.5

3.2.6 Summary and Conclusions

In this section, an off-line HTR system based on Hidden Markov Models and n -grams is presented. The HMM developed has a hierarchical structure with character models at the lowest level. These models are concatenated into words and into whole sentences. To incorporate linguistic information using n -grams, two different approaches have been studied here. The first one uses n -grams at word level, whereas on the second one the n -grams are used at character level. From the results, we can conclude that using n -grams at word level provide better information and so, the results are better. The obtained results can be considered as a baseline in the Computer Assisted Transcription of Text Images (CATTI) system and in the Multimodal CATTI (MM-CATTI) system that we are going to study in the next chapters.

3.3 On-line Handwritten Text Recognition

The on-line HTR system presented in this section, is used in this work to intend to decode the feedback touchscreen data for multimodal text correction on the MM-CATTI system; i.e. to recognize the pen strokes (words) written by the user in successive CATTI interactions in order to correct or replace the errors produced by the main, off-line HTR decoder. In general, touchscreen data consists of a series of pen-positions (x_t, y_t) , sampled at regular time instants $t = 1, 2, \dots$. Each sample of this “trajectory” can be accompanied by information about the

pen pressure, or at least by one bit indicating whether the pen is actually touching the screen or not. In this work no pressure information is used. The conceptual architecture adopted for the on-line HTR system is analogous to that used in the off-line HTR system, with exception of the preprocessing and feature extraction modules, which are explained hereafter.

3.3.1 Preprocessing

An overview of preprocessing techniques for on-line HTR can be seen in [HZK07]. In this work, the preprocessing of each trajectory involves only two simple steps: repeated points elimination and noise reduction.

Repeated points elimination: Repeated points appear in a trajectory when the pen remains down and motionless for some time. These uninformative data are trivially removed, along with the points marked as “*pen-up*”.

Noise reduction: Noise in pen strokes is due to erratic hand motion and inaccuracy of the digitalization process. To reduce this kind of noise, a simple smoothing technique is used which replaces every point (x_t, y_t) in the trajectory by the mean value of its neighbours [JMRW01].

Note that the temporal order of the data points is preserved throughout these preprocessing steps.

3.3.2 Feature Extraction

Each preprocessed trajectory is transformed into a new temporal sequence of 6-dimensional real-valued feature vectors [TPV07, MAE05]. The coordinate x is not used as a feature because of x range for different instances of the same character, can vary greatly depending on the position of the character into a word. The six features computed for each sample point are:

Normalized Vertical Position y_t : the coordinate pairs of each trajectory point are linearly scaled and translated to obtain new y_t values in the range $[0, 100]$, preserving the original aspect-ratio of the trajectory.

Normalized First Derivatives: x'_t and y'_t are calculated using the method given in [YOO+97]:

$$x'_t = \frac{\Delta x_t}{\|\nabla\|} \quad y'_t = \frac{\Delta y_t}{\|\nabla\|} \quad (3.8)$$

where,

$$\Delta x_t = \sum_{i=1}^r i \cdot (x_{t+i} - x_{t-i}) \quad \Delta y_t = \sum_{i=1}^r i \cdot (y_{t+i} - y_{t-i}) \quad \|\nabla\| = \sqrt{\Delta x_t^2 + \Delta y_t^2}$$

and r defines a window of size $2r + 1$ which determines the neighbour points involved in the computation. Setting $r = 2$ has provided satisfactory results in this work.

It is worth noting that the normalization of derivatives by $\|\Delta\|$ implicitly entails an effective *writing speed normalization*. In our experiments, this has proved to lead to better results than using explicit speed normalization preprocessing techniques such as *trace segmentation*, based on resampling the trajectory at equal-length (rather than equal time) intervals [PTV05, VLOK01].

Second derivatives: x_t'' and y_t'' , are computed in the same way as the first derivatives, but using x_t' and y_t' instead of x_t and y_t .

Curvature: k_t , is the inverse of the local radius of the trajectory in each point. It is calculated as:

$$k_t = \frac{x_t' \cdot y_t'' - x_t'' \cdot y_t'}{(x_t'^2 + y_t'^2)^{3/2}} \quad (3.9)$$

Although this feature is an explicit combination of the previous features, it has lead to slightly but consistently improved results in our experiments.

3.3.3 Recognition

Modelling and search for on-line recognition follows almost the same schemes used in off-line recognition, described in Section 3.2.3.

As in the off-line case, we use left-to-right continuous density character HMMs with a mixture of Gaussian densities assigned to each state. However, instead of using a fixed number of states for all HMMs, in this case it is variable for each character class. The number of states N_{Sc} chosen for each HMM character class M_c was computed as $N_{Sc} = l_c/f$, where l_c is the average length of the sequences of feature vectors used to train M_c , and f is a design parameter measuring the average number of feature vectors modelled per state (*state load factor*). This rule of setting up N_{Sc} tries to balance modelling effort across states. On the other hand, lexical modelling is carried out in exactly the same way as in the off-line HTR case.

Language modelling and search are simpler in this case because, as we will see in Section 5.4, we restrict our MM-CATTI study to single whole-word touchscreen corrections. That is, the language models used in the MM-CATTI search only allow one word per user-interaction.

3.3.4 Experimental Framework

On this section the experimental framework adopted to assess the accuracy of the HTR system presented in this section is described. The corpus used on the experiments, the assessment measures and the parameters to test are defined.

Corpora

The experiments carried out in order to test the effectiveness of the on-line HTR system studied here use the on-line UNIPEN corpus presented in the Section 2.5. As previously

discussed, the on-line handwritten words needed to assess the performance of the on-line HTR feedback subsystem used on the MM-CATTI system (see Chapter 5) were generated by concatenating random character instances from three UNIPEN categories: digits, lowercase letters and symbols.

In order to tune the parameters of the 68 on-line character HMMs needed, experiments were carried out on each of the 1a, 1c and 1d UNIPEN categories, partitioned into the training and test sets shown in the Table 2.6.

Assessment Measures

Since only single-character classification is considered, the conventional classification error rate (ER) will be used to assess the accuracy of the on-line HTR system. The ER is computed by comparing the character proposed by the system with the reference character, computing the percentage of characters misclassified.

Parameters

Two parameters must be adjusted in order to design an accurate recognizer. These parameters are the number of Gaussian densities and the *state load factor* (f) that measures the average number of feature vectors modelled per state.

3.3.5 Results

Different experiments have been carried out to assess the feasibility and potential of the on-line HTR system presented here.

All the samples were preprocessed using the preprocessing and feature extraction methods outlined in Sections 3.3.1 and 3.3.2. In order to tune the parameters of the 68 on-line character HMMs needed, experiments were carried out on each of the 1a, 1c and 1d UNIPEN categories, partitioned into the training and test sets shown in the Table 2.6. In this case 16 (diagonal) Gaussian densities were found to be optimal for the HMM state mixtures. On the other hand, the *state load factor* (f) was tuned through isolated character classification experiments, with best results obtained for $f = 10$. The classification error rates (ER) obtained for digits, letters and symbols were 1.7%, 5.9% and 21.8%, respectively. These results are comparable with those of the state-of-the-art obtained for this dataset. For example in [AKVGP04] classification error rates (ER) of 1.5% and 6% are reported for isolated digits and letters, respectively, by using Support Vector Machine. Moreover in [PLG01], employing neural networks, ER of 3% and 14% for digits and letters are presented. Finally in [Rat03], an online scanning n -tuple classifier system is used for classifying isolated digits, letters and symbols, obtaining in this case ER of 1.1%, 7.6% and 20.4% respectively.

3.3.6 Summary and Conclusions

Here, the on-line feedback HTR system that we are going to use in the MM-CATTI is presented. In the same way that the off-line HTR system presented in Section 3.2, this system is based on Hidden Markov Models and n -grams. The parameters values defined here will be used to train the HMMs on the Chapter 5.

Bibliography

- [aBJH93] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1993.
- [AF00] A. Amin and S. Ficher. A document skew detection method using the Hough transform. *Pattern Analysis and Applications*, 3:243–253, 2000.
- [AKVGP04] Abdul Rahim Ahmad, M. Khalia, C. Viard-Gaudin, and E. Poisson. Online handwriting recognition using support vector machine. In *TENCON 2004. 2004 IEEE Region 10 Conference*, volume A, pages 311–314, 21–24 2004.
- [BRKR00] A. Brakensiek, J. Rottland, A. Kosmala, and G. Rigoll. Off-Line Handwriting Recognition Using Various Hybrid Modeling Techniques and Character N-Grams. In *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 343–352, Amsterdam, The Netherlands, 2000.
- [BS89] R. Bozinovic and S. Srihari. Off-line cursive script word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):68–83, 1989.
- [BSM99] I. Bazzi, R. Schwartz, and J. Makhoul. An Omnifont Open-Vocabulary OCR System for English and Arabic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):495–504, 1999.
- [DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. J. Wiley and Sons, 1973.
- [DRI06] Fadoua DRIRA. Towards restoring historic documents degraded over time. In *Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 350–357, Washington, DC, USA, 2006. IEEE Computer Society.
- [EYSS99] A. El-Yacoubi, M. Guilloux, R. Sabourin, and C. Y. Suem. An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):752–760, 1999.
- [HZK07] B. Q. Huang, Y. B. Zhang, and M. T. Kechadi. Preprocessing techniques for online handwriting recognition. In *ISDA '07: Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*, pages 793–800, Washington, DC, USA, 2007. IEEE Computer Society.
- [iG07] Moisés Pastor i Gadea. *Aportaciones al reconocimiento automático de texto manuscrito*. PhD thesis, Departament de Sistemes Informàtics i Computació, València, Spain, Oct 2007. Advisors: E. Vidal and A.H. Tosselli.
- [Jel98] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

- [JMRW01] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. On-Line Handwriting Recognition: The NPen++ Recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–181, 2001.
- [Kat87] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35:400–401, March 1987.
- [KN95] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. volume 1, pages 181–184, Los Alamitos, CA, USA, 1995. IEEE Computer Society.
- [KS06] E. Kavallieratou and E. Stamatatos. Improving the quality of degraded document images. In *DIAL '06: Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 340–349, Washington, DC, USA, 2006. IEEE Computer Society.
- [KYWW82] R. G. Casey K. Y. Wong and F. M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, 1982.
- [MAE05] M.Pastor, A.H.Toselli, and E.Vidal. Writing speed normalization for on-line handwritten text recognition. In *Eighth International Conference on Document Analysis and Recognition (ICDAR05)*, volume II of *Lecture Notes in Computer Science*, pages 1131–1135. IEEE Computer Society, Seoul (Korea), August 2005.
- [MB01] U.-V. Marti and H. Bunke. Using a Statistical Language Model to improve the performance of an HMM-Based Cursive Handwriting Recognition System. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.
- [MMS99] F. Bortolozzi S. Garnes M. Morita, J. Facon and R. Saboruin. Mathematical morphology and weighted least squares to correct handwriting baseline skew. In *Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR '99)*, volume 1, pages 430–433, 1999.
- [OTI98] A. Ogawa, K. Takeda, and F. Itakura. Balancing acoustic and linguistic probabilities. In *Proceeding IEEE Conference Acoustics, Speech, and Signal Processing*, volume 1, 1998.
- [PLG01] M. Parizeau, A. Lemieux, and C. Gagn. Character Recognition Experiments using Unipen Data. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 481–485, 2001.
- [PS00] R. Plamondon and S. N. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

- [PTRV06] M. Pastor, A. H. Toselli, V. Romero, and E. Vidal. Improving handwritten off-line text slant correction. In *Procc. of The Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, Palma de Mallorca, Spain, August 2006.
- [PTV04] M. Pastor, A. Toselli, and E. Vidal. Projection profile based algorithm for slant removal. In *International Conference on Image Analysis and Recognition (ICIAR'04)*, Lecture Notes in Computer Science, pages 183–190, Porto, Portugal, September 2004. Springer-Verlag.
- [PTV05] M. Pastor, A. H. Toselli, and E. Vidal. Writing Speed Normalization for On-Line Handwritten Text Recognition. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR '05)*, pages 1131–1135, Seoul, Korea, August 2005.
- [Rab89] L. Rabiner. A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. *Proceedings IEEE*, 77:257–286, 1989.
- [Rat03] E. H. Ratzlaff. Methods, Report and Survey for the Comparison of Diverse Isolated Character Recognition Results on the UNIPEN Database. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR '03)*, volume 1, pages 623–628, Edinburgh, Scotland, August 2003.
- [RPTV06] V. Romero, M. Pastor, A. H. Toselli, and E. Vidal. Criteria for handwritten off-line text size normalization. In *Proceedings of The Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, Palma de Mallorca, Spain, August 2006.
- [SK83] D. Sankoff and J. B. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.
- [SR98] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.
- [TJK+04] A. H. Toselli, A. Juan, D. Keysers, J. González, I. Salvador, H. Ney, E. Vidal, and F. Casacuberta. Integrated Handwriting Recognition and Interpretation using Finite-State Models. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(4):519–539, June 2004.
- [TPV07] A. H. Toselli, M. Pastor, and E. Vidal. On-Line Handwriting Recognition System for Tamil Handwritten Characters. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *Lecture Notes in Computer Science (LNCS)*, pages 370–377. Springer-Verlag, Girona (Spain), June 2007.
- [Vin02] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1033–1446, 2002.

- [VLOK01] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Speeding Up On-line Recognition of Handwritten Characters by Pruning the Prototype Set. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR '01)*, pages 0501–0507, Seattle, Washington, September 2001.
- [YCL03] Shuhua Wang Yang Cao and Heng Li. Skew detection and correction in document images based on straight-line fitting. *Pattern Recognition Letters*, 24:1871–1879, 2003.
- [YOO+97] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book: Hidden Markov Models Toolkit V2.1*. Cambridge Research Laboratory Ltd, March 1997.
- [ZCB06] M. Zimmermann, J.-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):818–821, 2006. Member-Horst Bunke.

CHAPTER 4

Computer Assisted Transcription of Handwritten Text Images

4.1 Introduction

In this chapter the application of the IPR framework outlined in Section 1.3 about the transcription of handwritten documents is discussed. This application is called *Computer Assisted Transcription of Text Images (CATTI)* [TRPV09, TRRV07, RTRV07] and, rather than full automation, aims at assisting the humans in the proper recognition-transcription process; that is, facilitating and speeding up their task of transcription of handwritten texts. The new interactive, on-line framework, combines the efficiency of the automatic handwriting recognition system with the accuracy of the human transcriber, integrating the human activity into the recognition process and taking advantage of the user's feedback.

Figure 4.1 shows a schematic view of these ideas. Note, that the CATTI system presented follows the same architecture used on the Chapter 3, composed of four modules: preprocessing, feature extraction, training and recognition. The main difference is that the recognition module used here must be adapted to cope with the user feedback. Now, by observing the handwritten text image and the transcription hypothesis that the system derives from the image, the human transcriber must provide some feedback, which may interactively help the system to refine or to improve its hypothesis until it is finally accepted.

Note that the user provides feedback only to the recognition module. However, a more general user interaction approach would not necessarily be restricted only to this module. The user could interact directly with the preprocessing module in order to correct segmentation or preprocessing errors that the system take into account to improve the recognition accuracy.

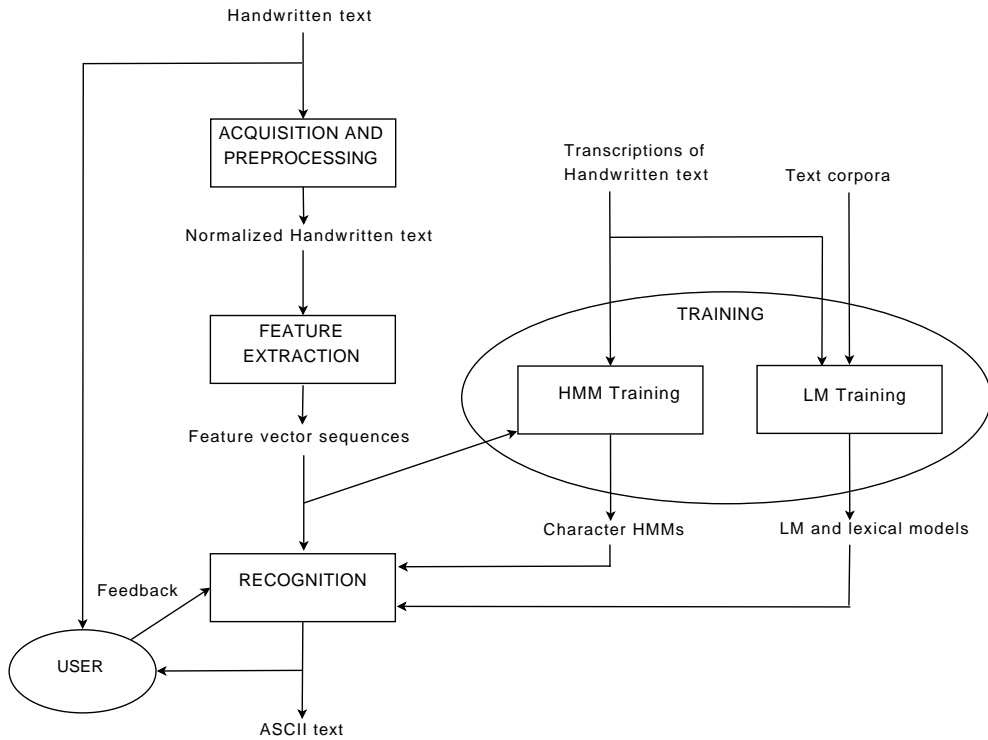


Figure 4.1: Overview of the CATTI system.

In this work, the detection and segmentation of text lines from each page has been carried out in a semi-supervised way, and we assume that it is error-free.

The process starts when the HTR system proposes a full transcription \hat{s} of a feature vector sequence \mathbf{x} , extracted from a handwritten text image (see Section 3.2). Then, the human transcriber (named user from now on) reads this transcription until he or she finds a mistake; i.e, he or she validates a prefix \mathbf{p}' of the transcription which is error-free. Now, the user can enter a word, v , to correct the erroneous text that follows the validated prefix. This action produces a new prefix \mathbf{p} (the previously validated prefix, \mathbf{p}' , plus the amendments introduced by the user, v). Then, the system takes into account the new prefix to suggest a suitable continuation to this prefix (i.e., a new \hat{s}), thereby starting a new cycle. This process is repeated until a correct, full transcription \mathbf{T} of \mathbf{x} is accepted by the user.

An example of this process is shown in Figure 4.2. It is worth noting in this example that non-interactive post-editing would have required the user to correct *five* errors from the original recognized hypothesis whereas, with the interaction feedback, only *two* user-corrections (the italic and underlined text in the final transcription \mathbf{T}) are necessary to get the final error-free transcription.

Next, the formal framework of CATTI will be explained on Sections 4.2, 4.3 and 4.4


	x							
INTER-0	p							
INTER-1	$\hat{s} \equiv \hat{w}$	antiguos	cuidadores	que	en	el Castillo	sus	llamadas
	p'	antiguos						
	<i>v</i>		ciudadanos					
INTER-2	p	antiguos	ciudadanos					
	\hat{s}	antiguos	ciudadanos	que	en	el Castillo	sus	llamadas
	p'	antiguos	ciudadanos	que	en			
FINAL	<i>v</i>					Castilla		
	p	antiguos	ciudadanos	que	en	Castilla		
	\hat{s}						se	llamaban
	<i>v</i>							#
	p \equiv T	antiguos	<u>ciudadanos</u>	que	en	<u>Castilla</u>	se	llamaban

Figure 4.2: Example of CATTI interaction to transcribe an image of the Spanish sentence “*antiguos ciudadanos que en Castilla se llamaban*”. Initially the prefix **p** is empty, and the system proposes a complete transcription $\hat{s} \equiv \hat{w}$ of the input image **x**. In each interaction step the user reads this transcription, accepting a prefix **p'** of it. Then, he or she types in some word, *v*, to correct the erroneous text that follows the validated prefix, thereby generating a new prefix **p** (the accepted one **p'** plus the word *v* added by the user). At this point, the system suggests a suitable continuation \hat{s} of this prefix **p** and this process is repeated until a complete and correct transcription of the input signal is reached. In the final transcription, **T**, the underlined italic words are the words typed by the user. In this example the estimated post-editing effort (WER) is 5/7 (71%), while the corresponding interactive estimate (WSR) is 2/7 (29%). This results in an estimated effort reduction (EER) of 59% (see Section 4.7.1 for definitions of WER, WSR and EER).

and some modifications to make the system more ergonomic and friendlier to the user will be explained on Sections 4.5 and 4.6. Then, some experiments are shown on Section 4.7. Finally, the conclusions are drawn in Section 4.9

4.2 Formal Framework

Formally, the CATTI framework can be seen as an instantiation of the problem formulated in Equation (1.2) where, in addition to the given image **x**, a user-validated *prefix* **p** of the transcription is available. This prefix, which corresponds to the feedback *f* in Equation (1.2), contains information from the previous system’s prediction plus user’s actions, in the form of amendment keystrokes. The HTR system should try to complete this prefix by searching for the most likely *suffix* \hat{s} (\hat{h} in Equation (1.2)), according to:

$$\hat{s} = \underset{s}{\operatorname{argmax}} \Pr(s \mid \mathbf{x}, \mathbf{p}) = \underset{s}{\operatorname{argmax}} \Pr(\mathbf{x} \mid \mathbf{p}, s) \cdot \Pr(s \mid \mathbf{p}) \approx \underset{s}{\operatorname{argmax}} P(\mathbf{x} \mid \mathbf{p}, s) \cdot P(s \mid \mathbf{p}) \quad (4.1)$$

Equation (4.1) is very similar to (3.3), being **w** the concatenation of **p** and **s**. As in Section 3.2.3, $P(\mathbf{x} \mid \mathbf{p}, s)$ can be approximated by HMMs and $P(s \mid \mathbf{p})$ by an *n*-gram model conditioned by **p**. The main difference is that now **p** is given. Therefore, the search must be

performed over all possible suffixes \mathbf{s} of \mathbf{p} and the language model probability $P(\mathbf{s}|\mathbf{p})$ must account for the words that can be written after the prefix \mathbf{p} .

In order to solve Equation (4.1), the signal \mathbf{x} can be considered split into two fragments, \mathbf{x}_1^b and \mathbf{x}_{b+1}^M , where M is the length of \mathbf{x} . By further considering the boundary point b as a hidden variable in Equation (4.1), we can write:

$$\begin{aligned}\hat{\mathbf{s}} &\approx \operatorname{argmax}_{\mathbf{s}} \sum_{1 \leq b \leq M} P(\mathbf{x}, b | \mathbf{p}, \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p}) \\ &= \operatorname{argmax}_{\mathbf{s}} \sum_{1 \leq b \leq M} P(\mathbf{x}_1^b, \mathbf{x}_{b+1}^M | \mathbf{p}, \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p})\end{aligned}\quad (4.2)$$

We can now make the naive (but realistic) assumption that \mathbf{x}_1^b does not depend on the suffix and \mathbf{x}_{b+1}^M does not depend on the prefix, to rewrite Equation (4.2) as:

$$\hat{\mathbf{s}} \approx \operatorname{argmax}_{\mathbf{s}} \sum_{1 \leq b \leq M} P(\mathbf{x}_1^b | \mathbf{p}) \cdot P(\mathbf{x}_{b+1}^M | \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p})\quad (4.3)$$

Finally, the sum over all the possible segmentations can be approximated by the dominating term, leading to:

$$\hat{\mathbf{s}} \approx \operatorname{argmax}_{\mathbf{s}} \max_{1 \leq b \leq M} P(\mathbf{x}_1^b | \mathbf{p}) \cdot P(\mathbf{x}_{b+1}^M | \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p})\quad (4.4)$$

This optimization problem entails finding an optimal boundary point, \hat{b} , associated with the optimal suffix decoding, $\hat{\mathbf{s}}$. That is, the signal \mathbf{x} is actually split into two segments, $\mathbf{x}_p = \mathbf{x}_1^{\hat{b}}$ and $\mathbf{x}_s = \mathbf{x}_{\hat{b}+1}^m$. Therefore, the search for the best transcription suffix that completes a prefix \mathbf{p} can be performed just over segments of the signal corresponding to the possible suffixes and, on the other hand, we can take advantage of the information coming from the prefix to tune the language model constraints modelled by $P(\mathbf{s} | \mathbf{p})$. $P(\mathbf{x}_1^b | \mathbf{p})$ and $P(\mathbf{x}_{b+1}^M | \mathbf{s})$ can be modelled by HMMs.

4.3 Adapting the Language Model

Perhaps the simplest way to deal with $P(\mathbf{s} | \mathbf{p})$ is to adapt an n -gram language model to cope with the consolidated prefix. Given that a conventional n -gram models the probability $P(\mathbf{w})$ (where \mathbf{w} is the concatenation of \mathbf{p} and \mathbf{s} , i.e the whole sentence), it is necessary to modify this model to take into account the conditional probability $P(\mathbf{s} | \mathbf{p})$.

As discussed in [RCV07], let $\mathbf{p} = \mathbf{w}_1^k$ be a consolidated prefix and $\mathbf{s} = \mathbf{w}_{k+1}^l$ be a possible suffix. We can compute $P(\mathbf{s} | \mathbf{p})$ as it is shown in Equation (4.5).

$$\begin{aligned}P(\mathbf{s} | \mathbf{p}) &= \frac{P(\mathbf{p}, \mathbf{s})}{P(\mathbf{p})} \\ &= \frac{\prod_{i=1}^l P(w_i | \mathbf{w}_{i-n+1}^{i-1})}{\prod_{i=1}^k P(w_i | \mathbf{w}_{i-n+1}^{i-1})} \\ &= \prod_{i=k+1}^l P(w_i | \mathbf{w}_{i-n+1}^{i-1})\end{aligned}\quad (4.5)$$

Moreover, for the terms from $k+1$ to $k+n-1$ of this factorization, we have additional information coming from the already known words w_{k-n+2}^k , allowing us to decompose Equation (4.5) as:

$$\begin{aligned}
 P(\mathbf{s} \mid \mathbf{p}) &= \prod_{i=k+1}^{k+n-1} P(w_i \mid \mathbf{w}_{i-n+1}^{i-1}) \cdot \prod_{i=k+n}^l P(w_i \mid \mathbf{w}_{i-n+1}^{i-1}) \\
 &= \prod_{j=1}^{n-1} P(s_j \mid \mathbf{p}_{k-n+1+j}^k, \mathbf{s}_1^{j-1}) \cdot \prod_{j=n}^{l-k} P(s_j \mid \mathbf{s}_{j-n+1}^{j-1}) \quad (4.6)
 \end{aligned}$$

The first term of Equation (4.6) accounts for the probability of the $n-1$ words of the suffix, whose probability is conditioned by words from the validated prefix, and the second one is the usual n -gram probability for the rest of the words in the suffix.

4.4 Searching

In the first iteration of the CATTI process, \mathbf{p} is empty. Therefore, the decoder has to generate a full transcription of \mathbf{x} as shown in Equation (3.3). Afterwards, the user-validated prefix \mathbf{p} has to be used to generate a suitable continuation \mathbf{s} in the following interactions of the interactive transcription process.

We can explicitly rely on Equation (4.4) to implement a decoding process in one step, as in conventional HTR systems. The decoder should be forced to *match* the previously validated prefix \mathbf{p} and then continue searching for a suffix $\hat{\mathbf{s}}$ according to the constraints (4.6) [RCV07].

In this section, two possible implementations of the CATTI decoder are described. The first one is based on the well known Viterbi algorithm [Jel98], and the other one is based on word-graph techniques (see Chapter 1) similar to those described in [BBC+09, LS06] for Computer Assisted Translation and for multimodal speech post-editing. The computational cost of the second one is much lower than using the naïve Viterbi adaptation, at the expense of a moderate accuracy degradation. Therefore, using word-graph techniques the system is able to interact with the human transcriber in a time-efficient way.

4.4.1 Viterbi-based approach

The search problem corresponding to Equations (4.4) and (4.6) can be solved by building a special language model which can be seen as the “concatenation” of a *linear* model which strictly accounts for the successive words in \mathbf{p} and a “suffix language model” of Equation (4.6). First, an n -gram is built from the available training set. Then, a linear model which accounts for the validated prefix is constructed. Finally, these two models are combined into a single model. An example of this combination is shown the Figure 4.3.

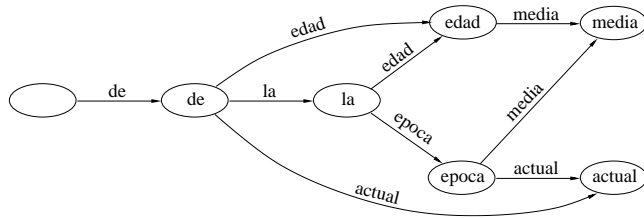
Owing to the finite-state nature of this special language model, the search involved in Equation (4.4) can be efficiently carried out using the Viterbi algorithm. Apart from the optimal suffix decoding, $\hat{\mathbf{s}}$, a correspondingly optimal segmentation of the \mathbf{x} is then obtained as a byproduct.

Training samples (L)

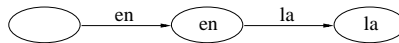
de la edad media
 de edad media
 de la epoca media
 de la epoca actual
 de la actual
 de actual

Prefix (L_p) = en la

Original Bigram (L)



Model for the Prefix (L_p)



Final Combined Model ($L_p L_s$)

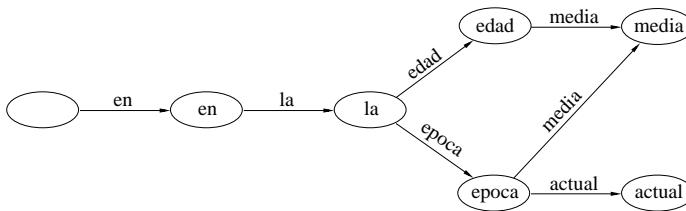


Figure 4.3: Example of a CATTI dynamic language model building. First, an n -gram (L) for the training set of the figure is built. Then, a linear model (L_p) which accounts for the prefix “en la” is constructed. Finally, these two models are combined into a single model ($L_p L_s$) as shown.

4.4.2 Word-graph based approach

It should be noted that the direct adaptation of the Viterbi algorithm, explained on the previous section, to implement these techniques leads to a computational cost that grows quadratically with the number of words of each sentence. This can be problematic for large sentences and/or for fine grained (character-level) interaction schemes. Nevertheless, using word-graph techniques very efficient, linear cost search can be easily achieved.

In this case, the search problem is solved using search techniques based on word graphs. As explained on Section 1.4.3, a word graph (WG) is a data structure that represents a set of strings in a very efficient way. In handwritten recognition, a WG represents the transcriptions with higher $\Pr(\mathbf{w} \mid \mathbf{x})$ of the given image text sentence. In this case, the word graph is just (a pruned version of) the Viterbi search trellis obtained when transcribing the whole image sentence. An example of a word graph is shown in Figure 1.5. This word graph represents a set of possible transcriptions of the Spanish sentence “antiguos ciudadanos que en Castilla se llamaban”.

As previously mentioned on Section 1.4.3 on Equations (1.7) and (1.8), the probability of

a word sequence, \mathbf{w} , in a word-graph is computed as the sum of the probabilities of all the paths that generate \mathbf{w} , $d(\mathbf{w})$:

$$P(\mathbf{w}) = \sum_{\phi_{\mathbf{w}} \in d(\mathbf{w})} \prod_{i=1}^l p(e_i) \quad (4.7)$$

where $\phi_{\mathbf{w}}$ is a sequence of edges e_1, e_2, \dots, e_l such that $\mathbf{w} = \omega(e_1), \omega(e_2), \dots, \omega(e_l)$. Given a WG, a word sequence with greatest probability can be written as:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{\phi_{\mathbf{w}} \in d(\mathbf{w})} \prod_{i=1}^l p(e_i) \quad (4.8)$$

However, as this maximization problem is NP-hard, we approximate it by means of the efficient Viterbi search algorithm:

$$P(\mathbf{w}) \approx \max_{\phi_{\mathbf{w}} \in d(\mathbf{w})} \prod_{i=1}^l p(e_i) \quad (4.9)$$

$$\hat{\mathbf{w}} \approx \underset{\mathbf{w}}{\operatorname{argmax}} \max_{\phi_{\mathbf{w}} \in d(\mathbf{w})} \prod_{i=1}^l p(e_i) \quad (4.10)$$

In handwritten recognition, the probability of an edge, $p(e)$, where $e = (i, j)$, is the product of the morphological lexical probability of the image element between its start and end node points $P(\mathbf{x}_{t(i)}^{t(j)} | \omega(e))$, times the language model probability of the given word at the edge $P(\omega(e))$. That is,

$$p(e) = P(\mathbf{x}_{t(i)}^{t(j)} | \omega(e)) \cdot P(\omega(e)) \quad (4.11)$$

In order to avoid the numeric underflow problems that can appear using probabilities we are going to use log-probabilities. So, the Equation (4.11) can be rewritten as follow:

$$\varphi(e) = \log P(\mathbf{x}_{t(i)}^{t(j)} | \omega(e)) + \log P(\omega(e)) \quad (4.12)$$

In addition, the simple multiplication is modified to balance the absolute values of both probabilities. The most common modification is to use the *grammar scale factor*, α (GSF), and the *word insertion penalty*, β (WIP), as it is used in Equation (3.4). So, now the score of each edge is computed as:

$$\varphi(e) = \log P(\mathbf{x}_{t(i)}^{t(j)} | \omega(e)) + \alpha \log P(\omega(e)) + \beta \quad (4.13)$$

Note, that Equations (4.12) and (4.13) are identical when $\alpha = 1$ and $\beta = 0$.

During the CATTI process the system has to make use of this word graph in order to complete the prefixes accepted by the human transcriber. In other words, the search problem consists in finding the target suffix \mathbf{s} that maximizes the posterior probability given a prefix \mathbf{p} as described in Equation (4.1).

To solve this problem, first the decoder parses the previously validated prefix \mathbf{p} over the WG. This parsing procedure will end defining a set of nodes Q_p corresponding to paths from the initial node whose associated word sequence is \mathbf{p} . Then, the decoder continues searching for the suffix \mathbf{s} from any of the nodes in Q_p that maximizes the posterior probability. Therefore, the boundary point b in Equations (4.2)-(4.4) is now restricted to values $t(q) \forall q \in Q_p$ and Equation (4.4) is now approximated as:

$$\hat{\mathbf{s}} = \operatorname{argmax}_{\mathbf{s}} \max_{q \in Q_p} P(\mathbf{x}_1^{t(q)} | p) \cdot P(\mathbf{x}_{t(q)+1}^M | \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p}) \quad (4.14)$$

This search problem can be efficiently carried out using dynamic programming. In order to make the process faster, first, we apply the Viterbi algorithm backwards from the final node to the initial one. In this way, we compute the best path and its probability from any node to the final node. Then, we look for the set of boundary nodes Q_p . Finally, we only have to multiply the probability computed from the initial node to any node $q \in Q_p$ times the probability from q to the final node (previously computed) and choose the node with maximum probability.

Error-correction parsing

The word graph is a representation of a large *subset* of the possible transcriptions for a source handwritten text image, where the number of possible transcriptions depends of the word graph density. So, it may happen that some prefixes given by the user can not be exactly found in the word graph. The solution is not to use \mathbf{p} but looking for the prefix \mathbf{p}_e , from all the possibles prefix on the word graph, that best match the given prefix. So, now the problem consist in looking for the suffix \mathbf{s} that maximizes the posterior probability and the prefix $\hat{\mathbf{p}}_e$ that best match the given prefix \mathbf{p} . This problem can be formulated as:

$$\begin{aligned} (\hat{\mathbf{p}}_e, \hat{\mathbf{s}}) &\approx \operatorname{argmax}_{\mathbf{p}_e, \mathbf{s}} P(\mathbf{p}_e, \mathbf{s} | \mathbf{x}, \mathbf{p}) \\ &= \operatorname{argmax}_{\mathbf{p}_e, \mathbf{s}} P(\mathbf{x} | \mathbf{p}, \mathbf{p}_e, \mathbf{s}) \cdot P(\mathbf{p}_e, \mathbf{s} | \mathbf{p}) \\ &= \operatorname{argmax}_{\mathbf{p}_e, \mathbf{s}} P(\mathbf{x} | \mathbf{p}, \mathbf{p}_e, \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p}, \mathbf{p}_e) \cdot P(\mathbf{p}_e | \mathbf{p}) \\ &= \operatorname{argmax}_{\mathbf{p}_e, \mathbf{s}} \sum_{q \in Q} P(\mathbf{x}, q | \mathbf{p}, \mathbf{p}_e, \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p}, \mathbf{p}_e) \cdot P(\mathbf{p}_e | \mathbf{p}) \end{aligned} \quad (4.15)$$

We can make the naive assumption that \mathbf{x} and \mathbf{s} do not depend of \mathbf{p} and only depend of \mathbf{p}_e to rewrite Equation (4.15) as:

$$(\hat{\mathbf{p}}_e, \hat{\mathbf{s}}) \approx \operatorname{argmax}_{\mathbf{p}_e, \mathbf{s}} \sum_{q \in Q} P(\mathbf{x}, q | \mathbf{p}_e, \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p}_e) \cdot P(\mathbf{p}_e | \mathbf{p}) \quad (4.16)$$

and following similar assumptions made on Equations (4.3) and (4.4) we can rewrite the previous equation as:

$$(\hat{\mathbf{p}}_e, \hat{\mathbf{s}}) \approx \underset{\mathbf{p}_e, \mathbf{s}}{\operatorname{argmax}} \max_{q \in Q} P(\mathbf{x}_1^{t(q)} | \mathbf{p}_e) \cdot P(\mathbf{x}_{t(q)+1}^M | \mathbf{s}) \cdot P(\mathbf{s} | \mathbf{p}_e) \cdot P(\mathbf{p}_e | \mathbf{p}) \quad (4.17)$$

where $P(\mathbf{p}_e | \mathbf{p})$ gives the probability of \mathbf{p}_e given \mathbf{p} and its value depends on the similarity between \mathbf{p}_e and \mathbf{p} .

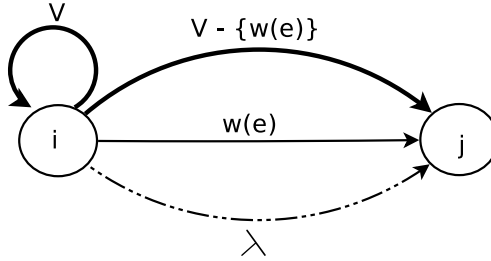


Figure 4.4: Example of edges added to a WG between the nodes i and j for probabilistic error correcting parsing. The edge labelled with the word $\omega(e)$ is the original edge and correspond to the operation of replacing the word $\omega(e)$ with itself. The group of edges labelled with $V - \{\omega(e)\}$ represent the substitution of $\omega(e)$ for another word. Here we have an edge for each word in the vocabulary except $\omega(e)$. The edge labelled with λ (empty symbol) models a deletion. Finally, the last group is for insertions, involving an edge for each word in the vocabulary from a state to itself.

$P(\mathbf{p}_e | \mathbf{p})$ can be modelled in terms of probabilistic error correcting parsing. This can be easily done by expanding the WG with a set of edges that represent the different editing operations. In Figure 4.4 we can see an example of all the added edges between two nodes [JE98].

Here the probabilities of the expanded edges are considered proportional to $\exp^{-d(v_1, v_2)}$, where $v_1, v_2 \in V \cup \lambda$ and $d(\cdot, \cdot)$ is the Levenstein distance between v_1 and v_2 . Until now, an edge was represented using its start and end node. However it is not possible now, because there is more than one edge between this two states. So, each edge must be represented using its start and end node and the word related with this edge, $e' = (i, j, v)$. Using log-probabilities the score of the different edges can be formulated as:

$$\varphi(i, j, v) = \begin{cases} \log P(\mathbf{x}_{t(i)}^{t(j)} | \omega(e)) + \alpha \log P(\omega(e)) + \beta - \gamma d(\omega(e), v) & i \neq j, \\ & v \in V \cup \{\lambda\} \\ \beta - \gamma d(\lambda, v) & i = j \end{cases} \quad (4.18)$$

where e is the original edge between the nodes i and j and the parameter γ weights the penalization due to the number of different characters. Its value has to be greater than 0 because, otherwise, we will be encouraging paths which are more different from the given

prefix. Note that if $\omega(e) = v$ the number of different characters will be 0, therefore the Equations (4.18) and (4.13) will become identical.

This heuristic can be implemented using dynamic programming and it can be further improved by visiting the states in WG in topological order [JE98], and incorporating beam-search techniques [Low76] to discard those states whose best score is worse than the best score at the current stage of the parsing multiplied by a given constant. Moreover, given the incremental nature of \mathbf{p} , the error-correcting algorithm takes advantage of this peculiarity to parse only the new suffix of \mathbf{p} provided by the user in the last interaction.

4.5 Increasing interaction ergonomy


	x							
INTER-0	\mathbf{p}	antiguos	cuidadores	que	en	el Castillo	sus	llamadas
INTER-1	\hat{s} <i>m</i>	antiguos	↑					
	\mathbf{p}'	antiguos						
INTER-2	\hat{s} <i>v</i>		cortesianos	que	en	el Castillo	sus	llamadas
	\mathbf{p}	antiguos	ciudadanos					
FINAL	\hat{s} <i>v</i>			que	en	Castilla	se	llamaban #
	$\mathbf{p} \equiv \mathbf{T}$	antiguos	<u>ciudadanos</u>	que	en	Castilla	se	llamaban

Figure 4.5: Example of CATTI operation. Starting with an initial recognized hypothesis \hat{s} , the user validates its longest well-recognized prefix \mathbf{p}' , making a mouse-action (*m*), and the system proposes a new suffix, \hat{s} . As the new hypothesis does not correct the mistake the user types the corrects word *v*, generating a new validated prefix \mathbf{p} (*v* concatenated to \mathbf{p}'). Taking into account the new prefix the system suggests a new hypothesis \hat{s} starting a new cycle. Now, the user validates the longest prefix \mathbf{p}' which is error-free. The system takes into account the new prefix \mathbf{p}' to propose a new suffix \hat{s} one more time. As the new hypothesis corrects the erroneous word a new cycle start. This process is repeated until the final error-free transcription \mathbf{T} is obtained. Underlined italic word in the final transcription is the only one which was corrected by the user. Note that in the iteration 1 it is needed a mouse-click to validate the longest prefix that is error-free and then, to type the correct word. However, the iteration 2 only needs a mouse-click.

In CATTI applications the user is repeatedly interacting with the system. Hence, making the interaction process easy is crucial for the success of the system. In conventional CATTI, before typing a new word in order to correct a hypothesis, the user needs to position the cursor in the place where she wants to type the word. This is done by performing a Mouse Action (MA) (or equivalent pointer-positioning keystrokes). By doing so, the user is already providing some very useful information to the system: he is validating a prefix up to the

position where he positioned the cursor, and, in addition, he is signalling that the following word located after the cursor is incorrect. Hence, the system can already take advantage of this fact and directly propose a new suitable suffix in which the first word is different to the first wrong word of the previous suffix. This way, many explicit user corrections are avoided [RTCV08, RTV09].

In Figure 4.5 we can see an example of the CATTI process with the new interaction mode. As in the conventional CATTI, the process starts when the HTR system proposes a full transcription \hat{s} of the input image \mathbf{x} . Then, the user reads this prediction until a transcription error is found (e) and makes a MA (m) to position the cursor at this point. This way, the user validates an error-free transcription prefix \mathbf{p}' . Now, before the user introduces a word to correct the erroneous one, the HTR system, taking into account the new prefix and the wrong word that follows the validated prefix, suggests a suitable continuation to this prefix (i.e., a new \hat{s}). If the new \hat{s} corrects the erroneous word (e) a new cycle starts. However, if the new \hat{s} has an error in the same position that the previous one, the user can enter a word, v , to correct the erroneous text e . This action produces a new prefix \mathbf{p} (the previously validated prefix, \mathbf{p}' , followed by v). Then, the HTR system takes into account the new prefix to suggest a new suffix and a new cycle starts. This process is repeated until a correct transcription of \mathbf{x} is accepted by the user.

Note that in the example shown in Figure 4.5, without interaction, a user should have to correct about *five* errors from the original recognized hypothesis. If the conventional CATTI is used the user only has to correct *two* words. However, with the new interaction mode only one user-correction is necessary to get the final error-free transcription. Note that in the iteration 1 a (single) MA does not succeeds and the correct word needs to be physically typed. However, the iteration 2 only needs a MA.

This new kind of interaction needs not be restricted to a single pointer-positioning MA. Several scenarios arise, depending on the number of times the user performs a MA. In the simplest one, the user only makes a MA when it is necessary to displace the cursor (single-MA). In this case the MA does not involve any extra human effort, because it is the same action that the user should make in the conventional CATTI to position the cursor before typing the correct word. Another scenario that can be considered consists in performing a MA systematically before writing, even in those cases where the cursor is already in the correct position. In this case, however, there is a cost associated to this kind of MAs, since the user does need to perform additional actions, which may or may not be beneficial. This scenario can be easily extended allowing to the user to make several MA before deciding to make an explicit word correction.

Since we have already dealt, in the Section 4.2, with the problem of finding a suitable suffix \hat{s} when the user validates a prefix \mathbf{p}' and introduces a correct word v , we focus now on the problem in which the user only makes a MA. In this case the decoder has to cope with the input image \mathbf{x} , the validated prefix \mathbf{p}' and the erroneous word that follows the validated prefix e , in order to search for a transcription suffix \hat{s} :

$$\hat{s} = \underset{s}{\operatorname{argmax}} \Pr(\mathbf{s} \mid \mathbf{x}, \mathbf{p}', e) \approx \underset{s}{\operatorname{argmax}} P(\mathbf{x} \mid \mathbf{p}', \mathbf{s}, e) \cdot P(\mathbf{s} \mid \mathbf{p}', e) \quad (4.19)$$

Similar assumptions and developments as those followed in Section 4.2 can be made to model $P(\mathbf{x} \mid \mathbf{p}', \mathbf{s}, e)$. On the other hand, $P(\mathbf{s} \mid \mathbf{p}', e)$ can be provided by a language model constrained by the validated prefix \mathbf{p}' and by the erroneous word that follows it.

4.5.1 Language Model and Search

$P(\mathbf{s} \mid \mathbf{p}', e)$ can be approached by adapting an n -gram language model so as to cope with the validated prefix \mathbf{p}' and with the erroneous word that follows it, e . The language model presented in Section 4.3 would provide a model for the probability $P(\mathbf{s} \mid \mathbf{p}')$, but now the first word of \mathbf{s} is conditioned by e . Therefore, some changes are needed.

Let $\mathbf{p}' = \mathbf{w}_1^k$ be the validated prefix and $\mathbf{s} = \mathbf{w}_{k+1}^l$ be a possible suffix. Considering that the wrongly-recognized word e only affects the first word of the suffix w_{k+1} , $P(\mathbf{s} \mid \mathbf{p}', e)$ can be computed as:

$$P(\mathbf{s} \mid \mathbf{p}', e) \simeq P(w_{k+1} \mid \mathbf{w}_{k+2-n}^k, e) \cdot \prod_{i=k+2}^{k+n-1} P(w_i \mid \mathbf{w}_{i-n+1}^{i-1}) \cdot \prod_{i=k+n}^l P(w_i \mid \mathbf{w}_{i-n+1}^{i-1}) \quad (4.20)$$

Now, taking into account that the first word of the possible suffix w_{k+1} has to be different to the erroneous word e , $P(w_{k+1} \mid \mathbf{w}_{k+2-n}^k, e)$ becomes:

$$P(w_{k+1} \mid \mathbf{w}_{k+2-n}^k, e) = \frac{\bar{\delta}(w_{k+1}, e) \cdot P(w_{k+1} \mid \mathbf{w}_{k+2-n}^k)}{\sum_{v'} \bar{\delta}(v', e) \cdot P(v' \mid \mathbf{w}_{k+2-n}^k)} \quad (4.21)$$

where $\bar{\delta}(v, v')$ is 0 when $v = v'$ and 1 otherwise. Since, $\sum_{v'} \bar{\delta}(v', e) \cdot P(v' \mid \mathbf{w}_{k+2-n}^k)$ is the same for any w_{k+1} , during the search process of $\hat{\mathbf{s}}$ (Equation (4.19)) we can approximate $P(w_{k+1} \mid \mathbf{w}_{k+2-n}^k, e)$ by $\bar{\delta}(w_{k+1}, e) \cdot P(w_{k+1} \mid \mathbf{w}_{k+2-n}^k)$.

As in the conventional CATTI the search problem involved in the Equation (4.19) can be solved by building a special language model, but now the ‘‘Suffix Language Model’’ of the Equation (4.20) is modified in accordance with Equation (4.21). Thanks to the finiteness of this special language model, the search involved in Equation (4.19) can be carried out using the Viterbi algorithm. However, an easier implementation can be carried out using word-graphs. The restrictions entailed by (4.21) can be easily implemented by deleting the edge labelled with the word e after the prefix has been matched. An example is shown in Figure 4.6. This example assumes the user has validated the prefix ‘‘antiguos ciudadanos que en’’ and the wrongly-recognized word was ‘‘el’’. Hence, the new word-graph has the edge labelled with the word ‘‘el’’ disabled.

4.6 CATTI at the character level

Until now, for the sake of clarity, human feedback for CATTI has been assumed to come in the form of whole-word interactions; i.e., the system does not start a new cycle until the user enters a whole-word. This allows us to properly compare the estimated user-effort reduction achieved by CATTI with respect to conventional post-editing of automatic transcriptions. Nevertheless, character-level keystroke interactions can allow for more ergonomic and friendly interfaces. In this section this new level of interaction is presented. Now, as soon as the user introduces a new keystroke, the system proposes a new suitable continuation.

In Figure 4.7 we can see an example of the CATTI process at character level. As in the conventional CATTI, the process starts when the HTR system proposes a full transcription $\hat{\mathbf{s}}$ of the input image \mathbf{x} . Then, the user validates the longest prefix that is error free \mathbf{p}' and

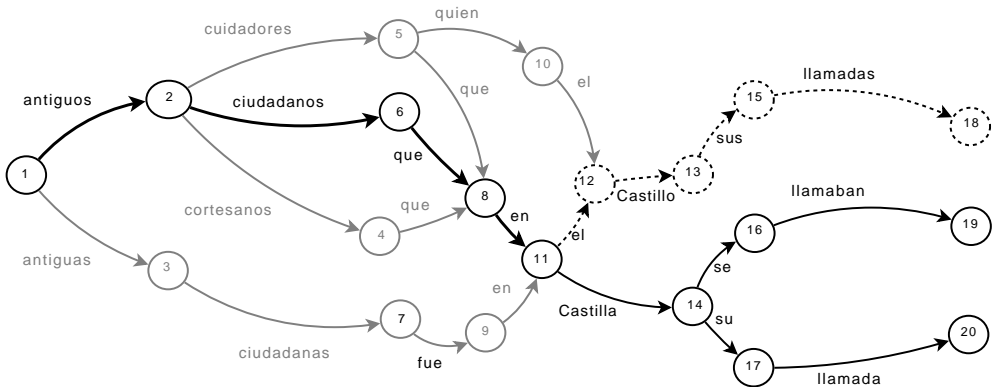


Figure 4.6: Example of word-graph generated after the user validates the prefix “antiguos ciudadanos que en”. The edge corresponding to the wrong-recognized word “el” was disabled.

enters a character c to correct the erroneous text that follows the validated prefix, producing a new prefix \mathbf{p} . The last word of this new prefix is not necessary complete, so, the system must suggest a suitable continuation, whose first part completes the incomplete word validated by the user. This process is repeated until a correct transcription of \mathbf{x} is accepted by the user.

	\mathbf{x}						
INTER-0	\mathbf{p}	antiguos	cuidadores	que	en	el Castillo	sus llamadas
INTER-1	$\hat{s} \equiv \hat{w}$	antiguos	c				
	\mathbf{p}'	antiguos	i				
	c	antiguos	ci				
INTER-2	\hat{s}	antiguos	udadanos	que	en	el Castillo	sus llamadas
	\mathbf{p}'	antiguos	ciudadanos	que	en		
	c	antiguos				C	
FINAL	\mathbf{p}	antiguos	ciudadanos	que	en	C	
	\hat{s}					astilla	se llamaban
	c						#
	$\mathbf{p} \equiv \mathbf{T}$	antiguos	<u>ciudadanos</u>	que	en	<u>Castilla</u>	se llamaban

Figure 4.7: Example of CATTI operation at character level. Starting with an initial recognized hypothesis \hat{s} , the user validates its longest well-recognized prefix \mathbf{p}' and corrects the following erroneous character c , generating a new validated prefix \mathbf{p} (c concatenated to \mathbf{p}'). This new prefix \mathbf{p} is submitted as additional help information to the recognition system, which based on this proposes a new suffix \hat{s} . This process goes on until the final error-free transcription \mathbf{T} is obtained. Underlined italic characters in the final transcription are those which were corrected by user.

Two different approaches to develop the character-level keystroke interaction can be carried out. The first one consists on using language models at character level. In this case a

degenerate lexicon model is used to describe the different characters that can be recognized during the recognition process, and the language model will account for the concatenation of characters into text lines or sentences. All the explanation carried out in previous sections has been made at word level. However, it can be extended at character level replacing \mathbf{w} with \mathbf{c} , where \mathbf{c} is a sequence of characters. In Chapter 3 we have carried out experiments using character language models and the obtained results shown that using this language models do not provide so information as the language models at word level. So, we have decided do not study this approach and using only word language models.

On the second approach we work with language models at word level exactly as we have done previously. In this case, the system looks for the most probable word that begins with the incomplete word that the user has validated. In order to “autocomplete” the last incomplete word of the prefix and propose a suitable continuation, we assume that the prefix \mathbf{p} is divided into two fragments: \mathbf{p}'' and v_p . \mathbf{p}'' is the part of the prefix formed by completed words and v_p is the last incomplete word of the prefix. In the example presented in Figure 4.7 in the first interaction, INTER-1, \mathbf{p}'' will be “antiguos” and v_p will be “ci”. In this case the decoder has to cope with the input image \mathbf{x} , the validated prefix \mathbf{p}'' and the incomplete word v_p , in order to search for a transcription suffix $\hat{\mathbf{s}}$, whose first part is the continuation of the incomplete word v_p :

$$\begin{aligned}\hat{\mathbf{s}} &= \underset{\mathbf{s}}{\operatorname{argmax}} \Pr(\mathbf{s} \mid \mathbf{x}, \mathbf{p}'', v_p) \\ &= \underset{\mathbf{s}}{\operatorname{argmax}} \Pr(\mathbf{x} \mid \mathbf{p}'', v_p, \mathbf{s}) \cdot \Pr(\mathbf{s} \mid \mathbf{p}'', v_p) \\ &\approx \underset{\mathbf{s}}{\operatorname{argmax}} P(\mathbf{x} \mid \mathbf{p}'', v_p, \mathbf{s}) \cdot P(\mathbf{s} \mid \mathbf{p}'', v_p)\end{aligned}\quad (4.22)$$

Assumptions and developments similar to those followed in Section 4.2 can be made here in order to model $P(\mathbf{x} \mid \mathbf{p}'', v_p, \mathbf{s})$. On the other hand, $P(\mathbf{s} \mid \mathbf{p}'', v_p)$ can be provided by a language model constrained by the part of the prefix formed by completed words \mathbf{p}'' and by the incomplete word that follows it v_p .

4.6.1 Language Model and Search

To model $P(\mathbf{s} \mid \mathbf{p}'', v_p)$ we assume that the suffix \mathbf{s} is divided into two fragments: v_s and \mathbf{s}' . v_s is the first part of the suffix that correspond with the final part of the incomplete word of the prefix, i.e. $v_p v_s = v$ where v is an existing word in the task dictionary, and \mathbf{s}' is the rest of the suffix. In the example shown in Figure 4.7, in the interaction 1 v_s is “udanos” and \mathbf{s}' is “que en el Castillo sus llamadas”. So, the search must be performed over all possible suffixes \mathbf{s} of \mathbf{p} , and the language model probability $P(v_s, \mathbf{s}' \mid \mathbf{p}'', v_p)$ must ensure that the concatenation of the last part of the prefix v_p , and the first part of the suffix, v_s , form an existing word (v) in the task dictionary. This probability can be decomposed into two terms:

$$P(v_s, \mathbf{s}' \mid \mathbf{p}'', v_p) = P(\mathbf{s}' \mid \mathbf{p}'', v_p, v_s) \cdot P(v_s \mid \mathbf{p}'', v_p) \quad (4.23)$$

the first term accounts for the probability of the whole-words in the suffix, and can be modelled with the language model presented in Section 4.3. The second term ensures that the first

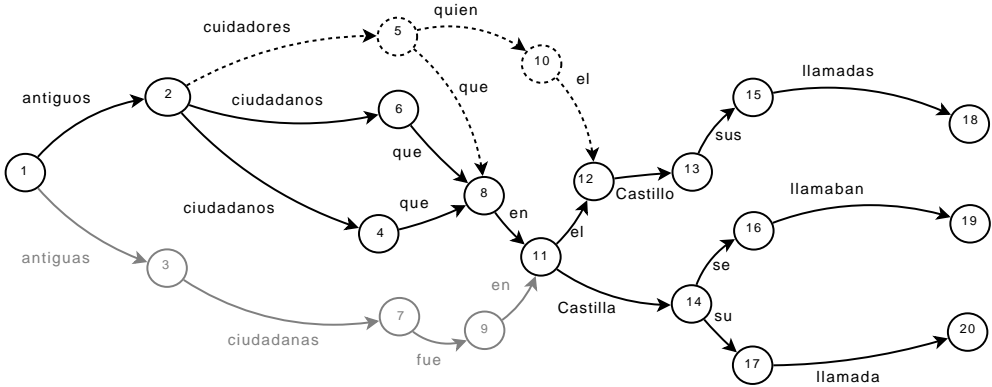


Figure 4.8: Example of word-graph generated after the user validates the prefix “antiguos ci” interacting at character-level. The edge whose word does not begin with “ci” are disabled.

part of the suffix v_s , will be a possible suffix of the incomplete word v_p , and can be written as:

$$P(v_s | \mathbf{p}'', v_p) = \frac{P(v_p, v_s | \mathbf{p}'')}{\sum_{v'_s} P(v_p, v'_s | \mathbf{p}'')} \quad (4.24)$$

To cope with the higher computational demands entailed by such a fine-grained operations, word graphs are used as in the conventional CATTI. The restrictions entailed by the Equation (4.24) can be easily implemented by deleting the edges labelled with a word that does not begin with v_p after the prefix has been matched. An example is shown in Figure 4.8. In this example the user has validated the prefix “antiguos ci”. Hence, in the new word-graph all the edges with a word that does not begin with “ci” after matching the prefix are disabled. If no edge is labelled with a word that begins with v_p , the system looks for the word in the word-graph vocabulary and then applies the error-correcting algorithm explained in Section 4.4.2. Finally, if no word in the word-graph vocabulary begins with v_p , the system looks for the word in the task vocabulary, applying then the error-correcting algorithm. This process will be called “CATTI autocompleting” from now on.

4.7 Experimental framework

The experimental framework adopted to assess the effectiveness of the CATTI system presented in this chapter is described in the following subsections. In addition some experiments in order to test the interaction at the character level and with mouse actions as additional information are defined.

The experiments have been performed using the three off-line corpora presented in Chapter 2 with the same assumptions made in Section 3.2.4.

4.7.1 Assessment Measures

Different evaluation measures have been adopted to assess the CATTI system. On the one hand, the quality of non-interactive transcription can be properly assessed with the well known *Word Error Rate* (WER). The WER is a good estimate of post-editing user effort.

On the other hand, the effort needed by a human transcriber to produce correct transcriptions using the CATTI system is estimated by the *Word Stroke Ratio* (WSR), which can be also computed using the reference transcription of the text image considered. The WSR can be defined as the number of (word level) user interactions that are necessary to achieve the reference transcriptions of the text images considered, divided by the total number of reference words.

This definition makes WSR and WER comparable. Moreover, the relative difference between them gives us a good estimate of the reduction in human effort that can be achieved by using CATTI with respect to using a conventional HTR system followed by human post-editing. This *Estimated Effort Reduction* will be denoted as “EFR”.

To assess the interaction method using mouse actions, we introduce the *Word Click Rate* (WCR). It can be defined as the number of additional mouse-clicks per word that the user has to do using the new user interaction mode. Note that the additional human effort needed for the verification of the transcription and positioning the cursor in the appropriate place in the conventional CATTI is the same as in the new single-MA user-CATTI interaction system. In both cases the user should read the transcription proposed by the system until he or she finds an error and then positions the cursor in the place where the new word has to be typed.

All the measures defined until now assess CATTI for whole-word interaction. However, character-level or keystroke interaction is also studied in this chapter. Therefore, some empirical measures must be introduced in order to compare the estimated user-effort reduction obtained by CATTI with respect to conventional post-editing at character level. On the one hand, the quality of the transcription without any system-user interactivity is given by the well known *Character Error Rate* (CER). It is defined as the minimum number of characters that need to be substituted, deleted or inserted to convert the sentences recognized by the system into the reference transcriptions, divided by the total number of characters in these transcriptions. The CER is a rough estimate of character-level post-editing user effort. In order to make the post-editing process more accurately comparable to the CATTI autocompleting approach, we introduce a “post-editing autocompleting” approach. Here, when the user enters a character to correct some incorrect word, the system automatically completes the word with the most probable word on the task vocabulary. Hence we define the *Post-editing Key Stroke Ratio* (PKSR), as the number of keystrokes that the user must enter to achieve the reference transcription, divided by the total number of reference characters.

On the other hand, the effort needed by a human transcriber to produce correct transcriptions using the CATTI system with autocompleting is estimated by the *Key Stroke Ratio* (KSR), which can be also computed using the reference transcriptions. The KSR can be defined as the number of (character level) user interactions that are necessary to achieve the reference transcription of the text image considered, divided by the total number of reference characters.

These definitions make PKSR and KSR comparable in a fair way. Moreover, the relative difference between them gives us a good estimate of the reduction in human effort that can

be expected by using CATTI with autocompleting with respect to using a conventional HTR system followed by human autocompleting post-editing (EFR). Note that EFR can also be used to compare CER with PKSR and CER with KSR. However, CER and KSR can not be compared in a fair way. For KSR we are using a system with the autocompleting, however it is not the case on the CER result. So if we compare the KSR with the CER in addition to the advantage obtained by the CATTI system we are achieving an extra advantage by the autocompleting approach.

4.7.2 Parameters

To obtain an accurate CATTI system some parameters need to be adjusted. Since, the CATTI system is based on HMMs, the parameters corresponding to the design of these HMMs need to be adjusted. However, in this chapter we are going to use the best HMMs obtained in the previous chapter for conventional HTR in each task.

On other hand, as explained on Section 4.4.2, when word-graph search is adopted an additional weighted component that penalizes the score of each edge must be introduced. So, the parameter γ (Error Correcting Penalty- ECP) that weights this penalization must be tuned. The values tested of this parameter were: 100, 200, 300, 400, 500 and 600.

4.8 Results

Here the results obtained with the different approaches proposed in this chapter are shown. First, we performed experiments using the Viterbi-based approach to make sure that the CATTI system presented here could be useful for the user and save human effort. Then, experiments were carried out using the word-graph based approach that, although it can lose some accuracy, incurs a much lower computational cost, allowing the user to interact with the system in real time. Then, results using MA in the CATTI interaction process are reported. Finally, some experiments at the character-level, that can allow for more ergonomic and friendly interfaces, are carried out.

Viterbi-based approach

In the experiments carried out here, we have used the same GSF and WIP values employed to obtain the baseline, non-interactive results presented in Section 3.2.5. Table 4.1 shows the estimated interactive human effort (WSR) required for each task, in comparison with the corresponding estimated post-editing effort (WER from Table 3.3). It also shows the estimated effort reduction (EFR), computed as the relative difference between WER and WSR.

According to these results, to produce 100 words of a correct transcription in the ODEC task, for example, a CATTI user should have to type only less than 20 words; the remaining 80 are automatically predicted by CATTI. That is to say, the CATTI user would save about 80% of the (typing and, in part thinking) effort needed to produce all the text manually. On the other hand, when interactive transcription is compared with post-editing, from every 100 (non-interactive) word errors, the CATTI user should have to interactively correct only less than 83. The remaining 17 errors would be automatically corrected by CATTI, thanks to the feedback information derived from other interactive corrections. It is important to remember

Table 4.1: Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR) using the Viterbi-base search. All results are percentages.

Corpus	WER	WSR	EFR
ODEC	22.9	18.9	17.5
IAMDB	25.3	21.1	16.6
CS-page	28.5	26.9	5.7
CS-book	33.5	32.1	4.2

here, that these results do not take into account the errors due to page segmentation into lines. This fact may affect optimistically to the comparison between producing all the text manually and use the CATTI system. However, it does not affect the comparison between the post-editing approach and the CATTI system, because, in both cases, it is necessary to previously correct the errors of page segmentation into lines.

The different performance figures achieved in the different tasks can be explained by quality differences in the original images and also by the relative lexicon sizes and bigram estimation robustness. The later is particularly problematic in the case of CS which, in addition, suffers from a segmentation into relatively short, syntactically meaningless lines, which further hinders the ability of the bigram language model to capture relevant contextual information.

On the other hand, it is interesting to realize that CATTI is more effective for lines or sentences that have several errors; clearly, if a sentence has just one (word) error, it *must* be interactively corrected by the user and the best CATTI can do is to keep the remaining text unchanged. Obviously, this is not guaranteed by Equation (4.1) and, in the worst case, a single word change made by the user may lead to more errors; that is, $WSR \geq WER$. To analyse this behaviour, Figure 4.9 presents WER, WSR and EFR values for increasing initial numbers of errors per sentence.

As expected, the estimated effort reduction increases with the number of errors per sentence, which clearly assess the ability of CATTI to correct more than one error per interaction step in sentences with several misrecognized words. Also, for sentences with a single error, CATTI does not help at all or is even worse than post-editing. Therefore, in practice, a good implementation of a CATTI user interface should allow the user to disable CATTI predictions when doing some (single-word) corrections.

Taking this into account, Table 4.2 shows the same results of Table 4.1, but excluding from the computation all the sentences with zero and one errors. As expected, the estimated effort reductions are better under this assumption.

Word-graph based approach

In Figure 4.10 we can see the WSR and the EFR obtained for each task using word-graphs search for different values of the parameter γ in comparison with the corresponding WER. The word-graphs used in the experiments were generated with the same GSF and WIP values

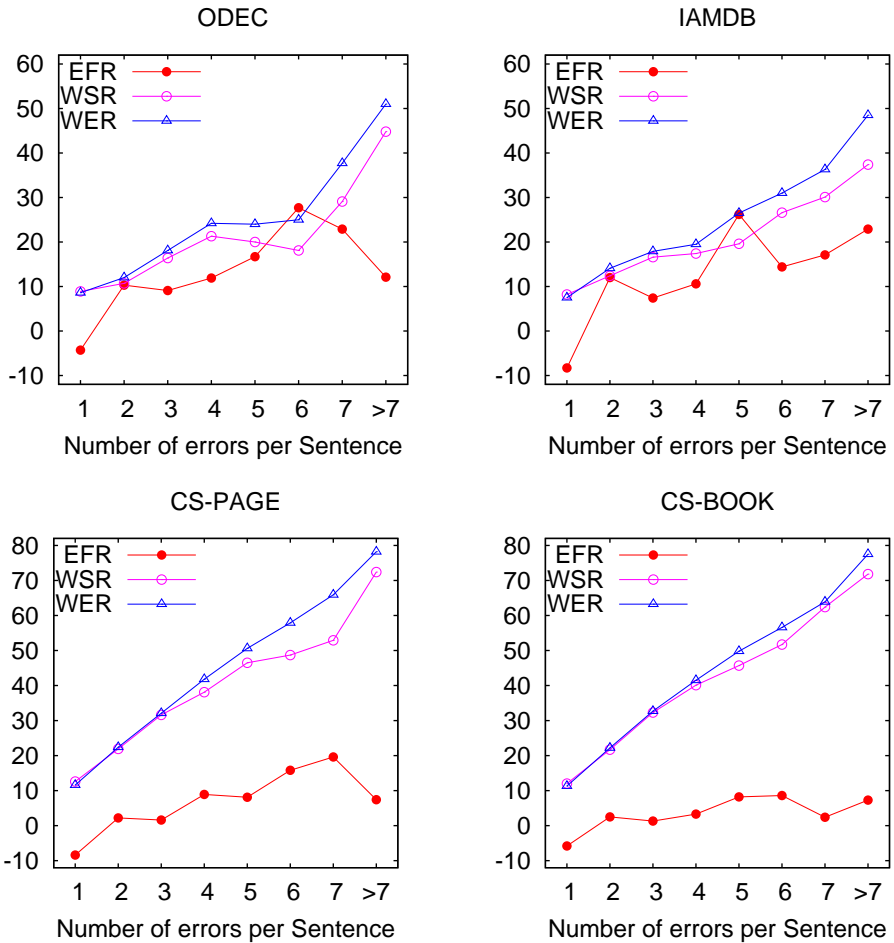


Figure 4.9: WER, WSR and EFR (all in %) for varying number of errors per sentence.

used for the baseline results. Table 4.3 summarizes the best WSR and EFR obtained for each task.

According to these results and just as we expected, the results obtained using the Viterbi based search are better than those obtained with word-graphs. This is owing to the fact that the word-graph is just a pruned version of the Viterbi search trellis. Therefore, not all the possible transcriptions for the input handwritten text image are available, leading to the loose of some system accuracy. However, the computational cost of using word-graphs is much lower than use the Viterbi adaptation allowing the human transcriber to interact with the system in real time.

Otherwise, it is clear from the results that the estimated human effort (EFR) to produce

Table 4.2: Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR), excluding the sentences with zero and one post-editing errors. All reported results are percentages.

Corpus	WER	WSR	EFR
ODEC-M3	30.7	25.2	17.9
IAMDB	30.2	24.6	18.4
CS-page	36.7	34.1	6.9
CS-book	42.0	40.0	4.8

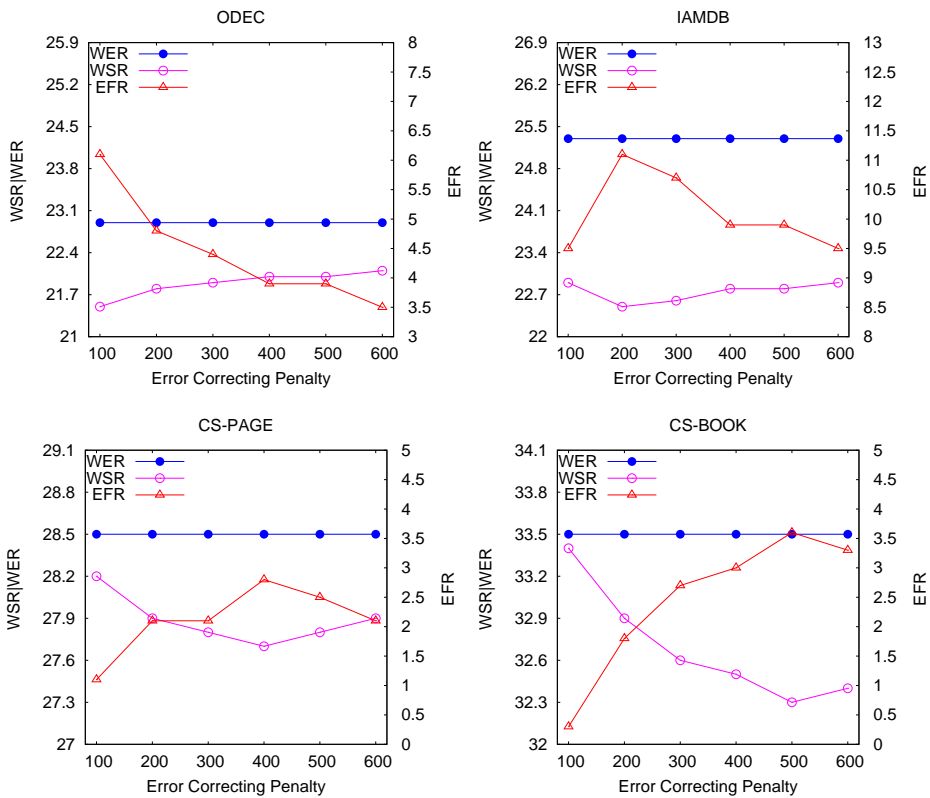


Figure 4.10: WER, WSR and EFR (all in %) for different values of the parameter γ .

error-free transcriptions with this CATTI approach is reduced in all the tasks.

As previously explained CATTI is more effective for lines or sentences that have several errors. Figure 4.11 presents WER, WSR and EFR values for increasing initial number of errors per sentence and the results are very similar to those obtained in Figure 4.9, showing

Table 4.3: Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR) using the word-graph based search. All results are percentages.

Corpus	WER	WSR	EFR
ODEC	22.9	21.5	6.1
IAMDB	25.3	22.5	11.1
CS-page	28.5	27.7	2.8
CS-book	33.5	32.3	3.6

that the EFR increases with the number of errors per sentence. Table 4.4 shows the same results of Table 4.2, but using word graphs. As happened using the Viterbi search, the EFR is better if we only take into account the sentences with more than one error.

Table 4.4: Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR), excluding the sentences with zero and one post-editing errors using word graphs. All reported results are percentages.

Corpus	WER	WSR	EFR
ODEC	30.7	28.6	6.8
IAMDB	30.2	26.3	12.9
CS-page	36.7	35.3	3.8
CS-book	42.0	40.3	4.1

Using MA in the CATTI interaction process

Owing to the easy implementation and the low computation cost, the new user-interaction with the CATTI system has been tested only in the word-graph approximation. Table 4.5 shows the results obtained with the new single-MA interaction mode (explained in Section 4.5). The first row shows the WSR obtained using the single-MA interaction mode. On the second row we can see the relative difference between the WSR obtained using the single-MA interaction mode with respect to the WSR obtained using the conventional CATTI (table 4.3). Finally, the last row shows the estimated effort reduction using the single-MA interaction mode with respect to using a conventional HTR system followed by human post-editing (WER of Table 4.3).

According to Table 4.5, the estimated human effort to produce error-free transcription using MA is significantly reduced with respect to using a conventional HTR system or the conventional CATTI (in both approximations). For example, in the IAMDB task, the new interaction mode can save about 26% of the overall effort, whereas the conventional CATTI would only save 11.1% using the word-graph approach, or 16.6% using the Viterbi search.

Figure 4.12 shows the WSR, the Estimated Effort-Reduction (EFR) with respect to WER

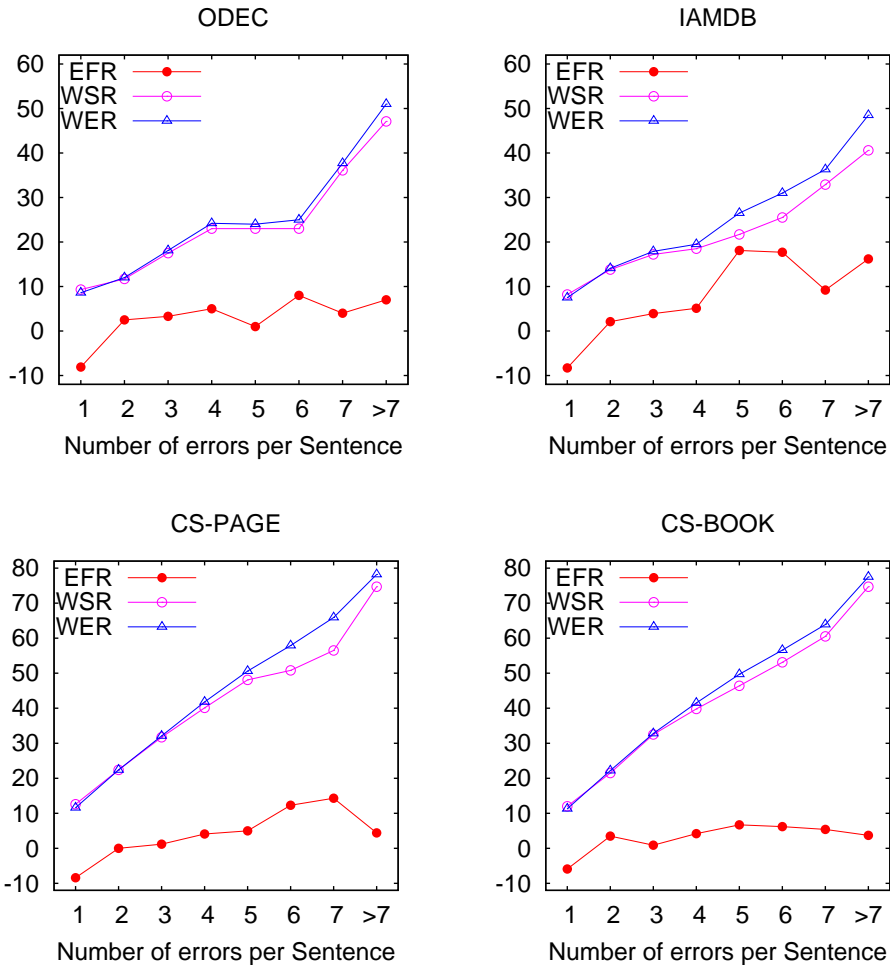


Figure 4.11: WER, WSR and EFR (all in %) for varying number of errors per sentence.

and the Word Click Rate (WCR) as a function of the maximal number of MA allowed by the user before writing the correct word. The first point (0) corresponds to the results of the conventional CATTI, and the point “S” corresponds to the the single-MA interaction considered in the previous table. A good trade-off is obtained when the maximum number of clicks is around 3, because a significant amount of expected human effort is saved with a fairly low number of extra clicks per word.

Table 4.5: Performance of the new single-MA interaction with CATTI system (WSR single-MA), along with Estimated Effort-Reduction for WSR single-MA with respect to WSR (EFR1) and WSR single-MA with respect to WER (EFR2). All results are percentages.

	ODEC	IAMDB	CS-PAGE	CS-BOOK
WSR single-MA	18.2	18.6	23.7	28.4
EFR1	15.3	17,3	14.4	12.1
EFR2	20.5	26.5	16.8	15.2

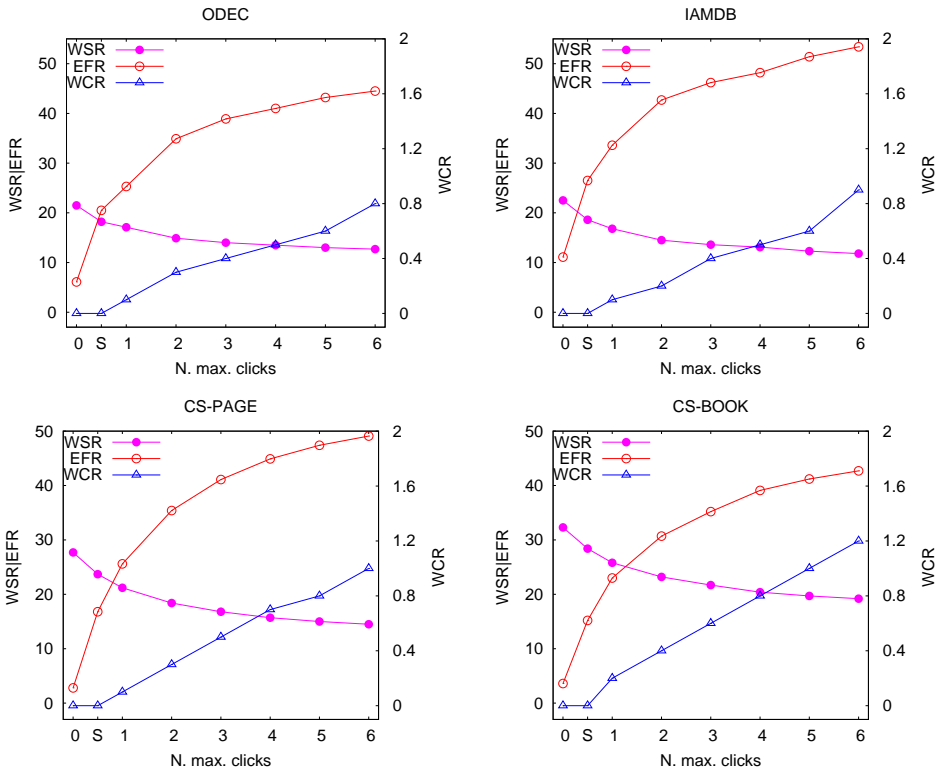


Figure 4.12: WSR, Estimated Effort-Reduction (EFR) and WCR as a function of the maximal number of MA allowed by the user before writing the correct word. The first point (0) correspond to the conventional CATTI, and the point *S* correspond to the single MA interaction discussed in Section 4.5.

CATTI at the character level

Table 4.6 shows the obtained results at character level on the three corpora, CS on its page and book partitions, ODEC and IAMDB, using the HTR system without any system-user

interactivity. We can see an estimation of the reduction in human effort (EFR) achieved by using the conventional HTR system with the auto-completing post-editing approach with respect to the plain HTR post-editing. Note that in the HTR system without interactivity, the auto-completing approach only consists in looking for the most probable word in the task vocabulary. That is, when the user corrects a character, the system automatically proposes a different word that begins with the given word prefix but, obviously, the rest of the sentence is not changed.

Table 4.6: CER and PKSR obtained with the post-editing auto-completing approach on the HTR system. EFR for PKSR with respect to CER is also shown. All results are percentages.

CORPUS	CER	PKSR	EFR
ODEC	12.8	9.0	29.7
IAMDB	14.8	13.5	8.8
CS-page	15.4	12.9	16.2
CS-book	18.2	15.5	14.8

Table 4.7 shows the obtained results using the CATTI system with auto-completing. On the column called “KSRvsCER” we can see the relative difference between using a CATTI system with auto-completing with respect to using the plain HTR post-editing system. However, as discussed in the previous section, this comparison is not very fair, because on the KSR we are using a system with auto-completing, whereas it is not the case on the CER result. On the other hand, the relative difference between KSR and PKSR, shown on the column called “KSRvsPKSR”, is totally fair, because in this case the two compared systems are working with the auto-completing approach. According to the results, the estimated human effort to produce error-free transcription using CATTI with the auto-completing approach is significantly reduced with respect to using the conventional HTR system with auto-completing post-editing. The new interaction level can save more than 20% of overall effort.

Table 4.7: KSR obtained with the CATTI auto-completing approach. EFR for KSR with respect to CER and KSR with respect to PKSR are also shown. All results are percentages.

CORPUS	KSR	EFR	
		KSRvsCER	KSRvsPKSR
ODEC	7.5	41.4	16.6
IAMDB	9.6	35.1	28.9
CS-page	10.5	31.8	18.6
CS-book	12.5	30.7	18.7

4.9 Conclusions and future work

In this chapter, we have proposed a new interactive, on-line framework, which combines the efficiency of automatic HTR systems with the accuracy of the paleography experts in the transcription of handwritten documents. In this proposal, the words corrected by the expert become part of a increasingly longer prefixes of the final target transcription. These prefixes are used by the CATTI system to suggest new suffixes that the expert can iteratively accept or modify until a satisfactory, correct target transcription is finally produced.

This system has been tested in three different tasks, ODEC, IAMDB and Cristo-Salvador. These tasks involve the transcription of handwritten answers from survey forms, handwritten full English sentences of different categories (editorial, religion, fiction, love, humour, ...) and an ancient handwritten document written in the year 1853, respectively. In spite of the extreme difficulty that entails the corpora used in the experiments, the obtained results are encouraging and show that the CATTI approach speed up the human error-correction process.

Two different implementations have been tested. The first one is based on the Viterbi algorithm, whereas the other one is based on word-graph techniques. From the obtained results we can conclude, that, although the results obtained using the Viterbi-based approach are better than the results using word-graphs, the word-graph approach is preferable. It is because the accuracy lost using word-graphs is not too high, while, the computational cost is much lower. This allows the human transcriber to interact with the system in real time.

In order to make the CATTI interaction process more comfortable, we have proposed a new way to interact with the CATTI system, by considering MAs as an additional information source: as soon as the user points to the next system error, the system proposes a new, hopefully more correct continuation. We have shown that this new user feedback can produce significant benefits, in terms of word stroke reductions. A simple implementation using word-graphs has been described and some experiments have been carried out. It is worth noting that alternative (n-best) suffixes could also be obtained with the conventional CATTI system. However, by considering the rejected words to propose the alternative suffixes, the interaction methods here studied are more effective and more comfortable for the user. Moreover, using the single-MA interaction method, a second alternative suffix is obtained without extra human effort.

In this chapter, character level interaction has been studied too. The interaction in conventional CATTI was in the form of whole-word interactions. In the new interaction level, as soon as the user introduces a new character the system proposes a new suitable suffix. A simple implementation of this new interaction level using word-graphs has been described and some experiments have been carried out. Considering the results obtained in the experiments, we can conclude that using this new interaction level not only allows for a more ergonomic and friendly interfaces, but a significant amounts of human effort in the handwritten text transcription process can be saved. For example, on the page partition of the CS corpus 18.6% of human effort can be saved, whereas using the CATTI system at word-level interaction the human estimated effort reduction was around 2.8%.

It is worth noting, however, that results obtained at the character and word levels are not directly comparable. A word-level correction encapsulates fairly well all the cognitive and physical human efforts needed to locate an error and type the correction. This is true both for off-line editing (WER) and for CATTI corrections (WSR) and therefore word-level EFR

figures can be considered quite fair. However, it is also clear that word-level corrections are less comfortable to users, in general. On the other hand, character-level corrections are much preferred by users, but it is unclear whether the number of keystrokes can be fairly used for assessment purposes. A corrective keystroke generally needs no significant cognitive effort since, in most cases, it is part of the correction of an already detected error. In other words, nor the CER neither the KSR account well for the cognitive component of corrective actions and it does not seem easy to establish a single, adequate scalar score that captures correctly the two kinds of human efforts involved at the character level. In the future we plan to carry out adequate field tests which will hopefully provide a more realistic assessment of the relative advantages of interacting at the character or at the word level.

Bibliography

- [BBC+09] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda H. Ney, J. Tomás, and E. Vidal. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.
- [JE98] J.C.Amengual and E.Vidal. Efficient Error-Correcting Viterbi Parsing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.PAMI-20, No.10:1109–1116, October 1998.
- [Jel98] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.
- [Low76] Bruce T. Lowerre. *The harpy speech recognition system*. PhD thesis, Carnegie mellon University, Pittsburgh, PA, USA, 1976.
- [LS06] P. Liu and F. K. Soong. Word graph based speech recognition error correction by handwriting input. In *ICMI '06: Proceedings of the 8th international conference on Multimodal Interfaces*, pages 339–346, New York, NY, USA, 2006. ACM.
- [RCV07] L. Rodríguez, F. Casacuberta, and E. Vidal. Computer Assisted Speech Transcription. In *Proceedings of the third Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *Lecture Notes in Computer Science*, pages 241–248. Girona (Spain), June 2007.
- [RTCv08] V. Romero, A. H. Toselli, J. Civera, and E. Vidal. Improvements in the computer assisted transcription system of handwritten text images. In *Proc. of the 8th Int. Workshop on Pattern Recognition in Information Systems (PRIS 2008)*, pages 103–112, Barcelona (Spain), June 2008.
- [RTRV07] V. Romero, A. H. Toselli, L. Rodríguez, and E. Vidal. Computer Assisted Transcription for Ancient Text Images. In *International Conference on Image Analysis and Recognition (ICIAR 2007)*, volume 4633 of *Lecture Notes in Computer Science*, pages 1182–1193. Springer-Verlag, Montreal (Canada), August 2007.
- [RTV09] V. Romero, A. H. Toselli, and E. Vidal. Using mouse feedback in computer assisted transcription of handwritten text images. In *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR)*, IEEE Computer Society. Barcelona, Spain, July 2009.
- [TRPV09] Alejandro Toselli, Verónica Romero, Moisés Pastor, and Enrique Vidal. Multimodal interactive transcription of text images. *Pattern Recognition. In Press, Accepted Manuscript*, 2009. doi = 10.1016/j.patcog.2009.11.019.
- [TRRV07] A. H. Toselli, V. Romero, L. Rodríguez, and E. Vidal. Computer Assisted Transcription of Handwritten Text. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 944–948. IEEE Computer Society, Curitiba, Paraná (Brazil), September 2007.

CHAPTER 5

Multimodal Computer Assisted Transcription of Handwritten Text Images

5.1 Introduction

Furthering the goal of making the interaction process friendlier to the user, led us to the development of *Multimodal CATTI* (MM-CATTI) [TRPV09, TRV08]. As discussed in Chapter 4, traditional peripherals like keyboard and mouse can be used to unambiguously provide the feedback associated with the validation and correction of the successive system predictions. Nevertheless, using more ergonomic multimodal interfaces should result in an easier and more comfortable human-machine interaction, at the expense of the feedback being less deterministic to the system. This is the idea underlying MM-CATTI, which focus on touch-screen communication, perhaps the most natural modality to provide the required feedback in CATTI systems. It is worth noting, however, that the use of this more ergonomic feedback modality comes at the cost of new, additional interaction steps needed to correct possible feedback decoding errors. Therefore, solving the multimodal interaction problem amounts to achieving a modality synergy where both main and feedback data streams help each-other to optimize overall performance.

As shown in Figure 1.1 (right) of the Chapter 1, the successive system's transcription hypotheses can be easily displayed on the touchscreen and user feedback corrections can be made through on-line pen-strokes and text which are exactly written over the text produced by the system.

On Figure 5.1 we can see a schematic view of the MM-CATTI system presented in this chapter. The main part of the system is the same presented in the Figure 4.1 for the CATTI

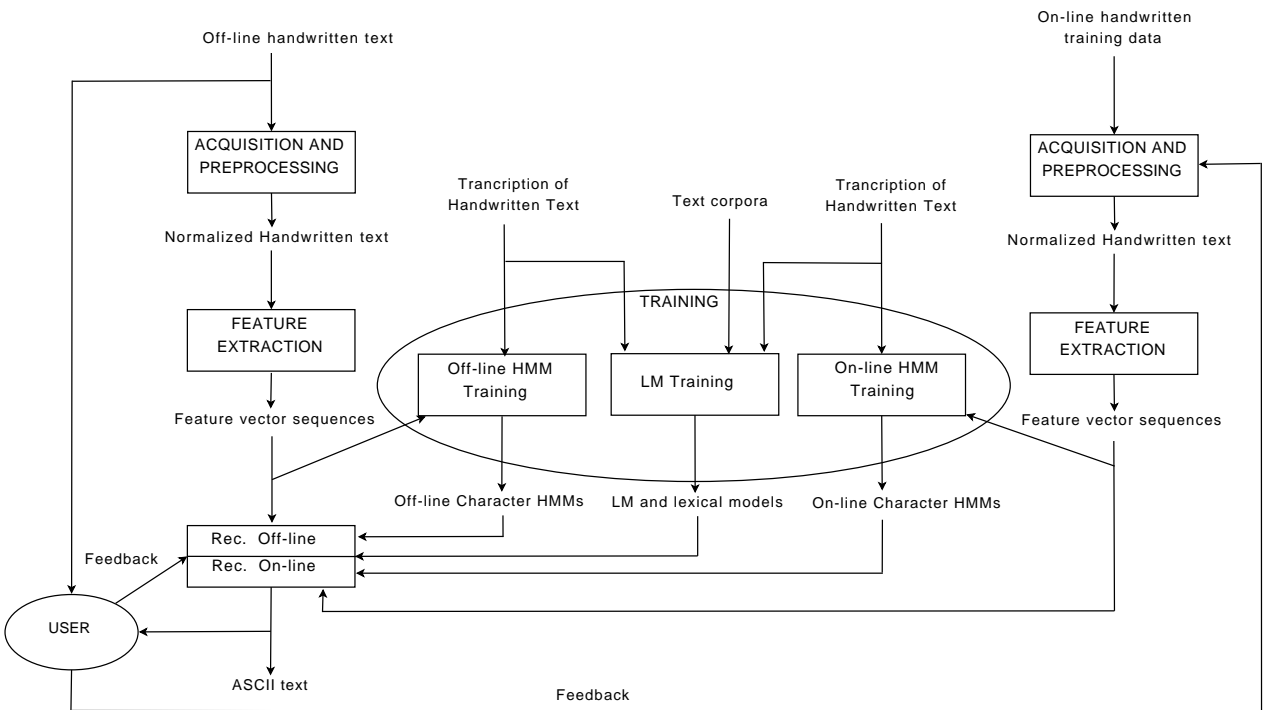


Figure 5.1: Overview of the MM-CATTI system.

system. However, since now the user feedback is provided in form of pen-strokes, an on-line HTR subsystem is introduced. This HTR subsystem follows the same architecture that the main HTR system and it is in charge to decode the feedback provided by the user. The feedback recognition module used here must be adapted to take advantage of interaction-derived information to boost its on-line HTR accuracy.

5.2 Formal Framework

More formally speaking, let \mathbf{x} be the input image and \mathbf{p}' the longest error-free prefix of the transcription. Let \mathbf{t} be the on-line *touchscreen pen strokes*^a provided by the user. These data are related to the suffix suggested by the system in the previous interaction step and are typically aimed at accepting or correcting parts of this suffix. Moreover, the user may additionally type some keystrokes (κ) on the keyboard in order to correct (other) parts of this suffix and/or to add more text. Using this information, the system has to suggest a new suffix, \mathbf{s} , as a continuation of the previous prefix \mathbf{p}' , the on-line touchscreen strokes \mathbf{t} and the typed text κ . That is, the problem is to find \mathbf{s} given \mathbf{x} and a feedback information composed of \mathbf{p}' , \mathbf{t} and κ , considering all possible *decodings*, d , of the on-line data \mathbf{t} (i.e., letting d be a hidden variable).

According to this very general discussion, it might be assumed that the user can type with independence of the result of the on-line handwritten decoding process. However, it can be argued that this generality is not realistically useful in practical situations. Alternatively, it is much more natural that the user waits for a specific system outcome (\hat{d}) from the on-line touchscreen interaction data (\mathbf{t}), prior to start typing amendments (κ) to the (remaining part of the previous) system hypothesis. Furthermore, this allows the user to fix possible on-line handwritten recognition errors in \hat{d} .

In this more pragmatic and simpler scenario, each interaction step can be formulated in two phases. In the first one, the system relies on the input image \mathbf{x} and the validated prefix of the previous interactive step \mathbf{p} , in order to search for a transcription suffix $\hat{\mathbf{s}}$:

$$\hat{\mathbf{s}} = \operatorname{argmax}_{\mathbf{s}} \Pr(\mathbf{s} \mid \mathbf{x}, \mathbf{p}) \quad (5.1)$$

Once $\hat{\mathbf{s}}$ is available the user validates the longest prefix, \mathbf{p}' , which is error free and produces some (may be null) on-line touchscreen data, \mathbf{t} . In the second phase, the system has to decode \mathbf{t} into a word, \hat{d} :

$$\hat{d} = \operatorname{argmax}_{d} \Pr(d \mid \mathbf{x}, \mathbf{p}', \hat{\mathbf{s}}, \mathbf{t}) \quad (5.2)$$

This action produces a new prefix \mathbf{p} , the previously validated prefix \mathbf{p}' , plus \hat{d} . If \hat{d} is not correct, the user can enter adequate amendment keystrokes κ , and produce a new consolidated prefix, \mathbf{p} , based on the previous \mathbf{p}' , \hat{d} and κ . The process continues in this way until \mathbf{p} is accepted by the user as a full correct transcription of \mathbf{x} .

An example of this kind of inter-leaved off-line image recognition and on-line touchscreen interaction is shown in Figure 5.2. In this example, we are assuming that on-line

^a*Pen strokes* are assumed to be sequences of real-valued vectors. See Section 3.3 for details.


	x						
INTER-0	p						
INTER-1	$\hat{s} \equiv \tilde{w}$	antiguos	cuidadores	que	en el Castillo	sus	llamadas
	\mathbf{p}' , t	antiguos	ciudadanos				
	\hat{d}		ciudadanas				
INTER-2	κ		OS				
	p	antiguos	ciudadanos				
	\hat{s}			que	en el Castillo	sus	llamadas
FINAL	\mathbf{p}' , t	antiguos	ciudadanos	que	en Castilla		
	\hat{d}				Castilla		
	κ	antiguos	ciudadanos	que	en Castilla		
	\hat{s}					se	llamaban
	κ						#
	p \equiv T	antiguos	<u>ciudadanos</u>	que	en <u>Castilla</u>	se	llamaban

Figure 5.2: Example of multimodal CATTI interaction with a CATTI system, to transcribe an image of the Spanish sentence “*antiguos ciudadanos que en Castilla se llamaban*”. Each interaction step starts with a transcription prefix **p** that has been fixed in the previous step. First, the system suggests a suffix \hat{s} and the user handwrites some touchscreen text, **t**, to amend \hat{s} . This defines a correct prefix \mathbf{p}' , which can be used by the on-line HTR subsystem to obtain a decoding of **t**. After observing this decoding, \hat{d} , the user may type additional keystrokes, κ , to correct possible errors in \hat{d} (and perhaps to amend other parts of \hat{s}). A new prefix, **p**, is built from the previous correct prefix \mathbf{p}' , the decoded on-line handwritten text, \hat{d} , and the typed text κ . The process ends when the user enters the special character “#”. System suggestions are printed in boldface and typed text in typewriter font. User corrections are shown in red. In the final transcription, **T**, typed text is additionally underlined. Assuming all interactions as whole-word corrections, the post editing WER would be 5/7 (71%), while the MM-CATTI WSR is 3/7 (43%); i.e., 2 touch-screen + 1 keyboard word corrections.

handwriting is the modality preferred by the user to make corrections, relying on the keyboard mainly (or only) to correct eventual on-line text decoding errors. Note that the potential increase in comfort of this setting comes at expense of a hopefully small number of additional interaction steps using the keyboard. In this example the user would need *three* interactive corrections using MM-CATTI, compared with the two keyboard-only corrections using CATTI and with the *five* post-editing word corrections required by the original, off-line recognized hypothesis.

Since we have already dealt with Equation (5.1) in Chapter 4 (Equation (4.1)-(4.4)), we focus now on Equation (5.2). As compared with Equation (1.2), here the triplet $(\mathbf{x}, \mathbf{p}', \hat{s})$ and **t** would correspond to the two modalities x and f , respectively. Therefore, assumptions and developments similar to those of Equation (1.3)-(1.4) lead to:

$$\hat{d} \approx \operatorname{argmax}_d P(d \mid \mathbf{x}, \mathbf{p}', \hat{s}) \cdot P(\mathbf{t} \mid d) \quad (5.3)$$

As in Chapter 4, $P(\mathbf{t} \mid d)$ is provided by (HMM) morphological models of the word in d (see Section 3.3.3 for details). On the other hand, here, $P(d \mid \mathbf{x}, \mathbf{p}', \hat{s})$ can be provided by a language model constrained by information derived from the input image \mathbf{x} , the

previous prefix \mathbf{p}' and by the suffix \hat{s} produced at the beginning of the current iteration. Equation (5.3) may lead to several scenarios depending on the assumptions and constraints adopted for $P(d \mid \mathbf{x}, \mathbf{p}', \hat{s})$. We examine some of them hereafter.

The simplest one corresponds to a conventional, non-interactive on-line HTR setting, where all the available conditions are ignored; i.e., $P(d \mid \mathbf{x}, \mathbf{p}', \hat{s}) \equiv P(d)$. This scenario is considered here as a *baseline*.

A more informative setting arises by taking into account information derived from the previous off-line HTR prediction \hat{s} . The user introduces the touchscreen data \mathbf{t} in order to correct the wrong word ($e = \hat{s}_1$) that follows the validated prefix \mathbf{p}' . Therefore, we can assume an *error-conditioned* model such as $P(d \mid \mathbf{x}, \mathbf{p}', \hat{s}) \equiv P(d \mid e)$; clearly, knowing the word that the user has already deemed incorrect should prevent the on-line decoder making the same error.

If, in addition to e , the information derived by the accepted prefix is also taken into account, a particularly useful scenario arises. In this case the decoding of \mathbf{t} is further constrained to be a suitable continuation of the prefix accepted so far, \mathbf{p}' ; that is:

$$P(d \mid \mathbf{x}, \mathbf{p}', \hat{s}) \equiv P(d \mid \mathbf{p}', e)$$

This model [TRV08] is the one studied in more detail here.

Finally, the most informative scenario corresponds to additionally using the information of the input image, as in Equation (5.3). In this case the on-line HTR decoder should find suitable continuations that are also good partial off-line transcriptions of the input text image. This potential source of increased performance is left for future studies.

5.3 Adapting the Language Model

Language modelling needed for the on-line HTR feedback subsystem in MM-CATTI is essentially similar to that described in Section 4.3 for the main, off-line HTR system. Language model constraints are implemented on the base of n-grams, depending on each multimodal scenario considered.

The simplest *baseline* scenario does not take into account any interaction-derived information and $P(d)$ could be provided by the same n-gram used for the off-line decoder. However, since only single whole-word touchscreen corrections are assumed, only uni-grams actually make sense.

The single-word assumption also simplifies the *error-conditioned* language model, $P(d \mid e)$ as follows:

$$P(d \mid e) = \begin{cases} 0 & d = e \\ \frac{P(d)}{1 - P(e)} & d \neq e \end{cases} \quad (5.4)$$

Finally, in MM-CATTI the language model probability is approximated by $P(d \mid \mathbf{p}', e)$. That is, the on-line HTR subsystem should produce a hypothesis \hat{d} for the touchscreen strokes \mathbf{t} , taking in account a user-accepted prefix, \mathbf{p}' , and the first wrong word, e , in the off-line HTR

suggestion. In this case, arguments similar to those in Section 4.3 apply and, under the same single whole-word assumption, we can use Equation (4.6) changing s with d , leading to:

$$P(d | \mathbf{p}', e) = \begin{cases} 0 & d = e \\ \frac{P(d | \mathbf{p}'_{k-n+2}^k)}{1 - P(e | \mathbf{p}'_{k-n+2}^k)} & d \neq e \end{cases} \quad (5.5)$$

where k is the length of \mathbf{p}' .

5.4 Searching

A simple implementation of Equation (5.5) is shown in Figure 5.3. In this example, $\mathbf{p}' = \text{“de la”}$ and the user wants to correct the wrong off-line recognized word “media”, by handwriting the word “edad” (for example) on the touchscreen. If the on-line HTR sub-system uses a bigram model, it is conditioned by the context word “la” (which is now the initial state) and the word transition edge “media” is disabled.

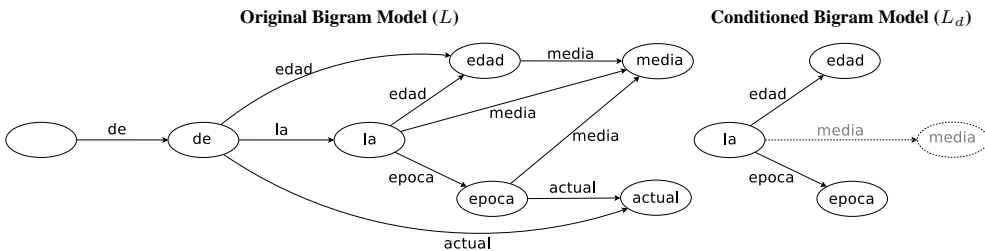


Figure 5.3: Example of MM-CATTI dynamic bigram language model generation. L is the original bigram model used by off-line HTR system, whereas L_d is the bigram sub-model, derived from L , which takes as initial state that corresponding to the prefix “la”. This simplified language model is used by the on-line HTR sub-system to recognize the touchscreen handwritten word “edad”, intended to replace the wrong off-line recognized word “media”, which was disabled in L_d .

As shown in the example, and unlike it happened in CATTI, the linear language model of the prefix \mathbf{p}' is no longer required, because the corresponding on-line touchscreen data of the prefix \mathbf{p}' do not exist in this case. Moreover, as we are assuming only single whole-word corrections, only the direct transitions from the starting node (the “la” node in the example) need to be considered.

As in CATTI searching (Section 4.4), owing to the finite-state nature of the n-gram language model, the search involved in Equation (5.5) can be efficiently carried out using the Viterbi algorithm.

5.5 Experimental Framework

The experimental framework adopted to assess the effectiveness of the MM-CATTI system presented in this chapter is described in the following subsections.

5.5.1 Corpora

The experiments have been performed on the three off-line corpora presented in Chapter 2 with the same assumptions made in Section 3.2.4. To train the on-line HTR feedback subsystem and test the MM-CATTI approach, the on-line handwriting UNIPEN corpus was chosen. As explained in Section 2.5, the UNIPEN data come organized into several categories such as lower and uppercase letters, digits symbols, isolated words and full sentences. Unfortunately, the isolated words category does not contain all (or almost none of) the required word instances that would have to be handwritten by the user in the MM-CATTI interaction process with the ODEC, IAMDB, or CS text images. Therefore, these words were generated by concatenating random character instances from three UNIPEN categories: *1a* (digits), *1c* (lowercase letters) and *1d* (symbols).

To increase realism, the generation of each of these test words was carried out employing characters belonging to the same writer. Three arbitrary writers were chosen, taking care that sufficient samples of all the characters needed for the generation of the required word instances were available from each writer. Each character needed to generate a given word was plainly aligned along a common word baseline, except if it had a descender, in which case the character baseline was raised 1/3 of its height. The horizontal separation between characters was randomly selected from *one* to *three* trajectory points. The selected writers are identified by their name initials as BS, BH and BR. Figure 5.4 shows examples of on-line word samples generated in this way, along real samples of the same words written by two writers in our labs.

Words from concatenated UNIPEN chars			Real word writing	
prendas	prendas	prendas	prendas	prendas
while	while	while	while	while

Figure 5.4: Examples of words generated using characters from the three selected UNIPEN test writers (BH, BR, BS), along with samples of the same words written by two other writers in our labs.

Training data were produced in a similar way using 17 different UNIPEN writers. For each of these writers, a sample of each of the 42 symbols and digits needed was randomly selected and one sample of each of the 1 000 most frequent Spanish and English words was generated, resulting in 34 714 training tokens (714 isolated characters plus 34 000 generated words). To generate these tokens, 186 881 UNIPEN character instances were used, using as many repetitions as required out of the 17 177 *unique* character samples available. Table 5.1 summarizes the amount of UNIPEN training and test data used in our experiments.

Table 5.1: Basic statistics of the UNIPEN training and test data used in the experiments.

Number of different:	Train	Test	Lexicon
writers	17	3	-
digits (1a)	1 301	234	10
letters (1c)	12 298	2 771	26
symbols (1d)	3 578	3 317	32
total characters	17 177	6 322	68

5.5.2 Assessment Measures

In order to assess the MM-CATTI system accuracy, different evaluation measures have been adopted. As in Chapter 4 the WER and the WSR are used to assess the quality of non-interactive transcriptions and the CATTI system respectively. The WSR of the MM-CATTI system will be decomposed into TS (touchscreen) and KBD (Keyboard). TS represents the percentage of corrections successfully made through the on-line HTR feedback modality. KBD is the percentage of corrections for which the feedback decoder failed and the correction had to be entered by means of the keyboard. On the other hand, as in CATTI, the EFR will give us a good estimate of the reduction in human effort that can be achieved by using MM-CATTI with respect to using a conventional HTR system followed by human post-editing.

Finally, since only single-word corrections are considered, the conventional classification error rate (ER) will be used to assess the accuracy of the on-line HTR feedback subsystem under the different constraints entailed by the MM-CATTI interaction process.

5.6 Results

The aim of these experiments is to assess the effectiveness of MM-CATTI in the scenarios described in Section 5.2. Multimodal operation offers ergonomics and increased usability at the expense of the system having to deal with non-deterministic feedback signals. Therefore, the main concern here is the accuracy of the on-line HTR feedback decoding and the experiments aim to determine how much this accuracy can be boosted by taking into account information derived from the proper interaction process. Ultimately, experiments aim at assessing which degree of synergy can actually be expected by taking into account both interactivity and multimodality.

In order to establish a word decoding baseline accuracy for the on-line HTR feedback subsystem, a simple word recognition experiment was carried out. As discussed in Section 5.5.1, the words needed to train and test the feedback subsystem for each task were generated by concatenating adequate UNIPEN characters. Therefore, new character HMMs were trained from these training words, using the parameters previously tuned through the isolated character recognition experiments (see Section 3.3.5). On the other hand, since only single words are to be recognized, a uni-gram language model was trained for each off-line task (CS-

page, CS-book, ODEC and IAMDB) to estimate the corresponding prior word probabilities. Table 5.2 shows the basic statistics of the data used in this experiment, along with the ER achieved by the non-interactive on-line HTR subsystem in the different tasks, using GSF values optimized for each language model. The words used as feedback correspond to the words that the user must to introduce during the CATTI process using an implementation based on the Viterbi algorithm. Similar experiments could be carried out using the words that the user must to introduce using an implementation based on word-graphs. In fact, the demonstrator presented on Chapter 6.3 has been implement using the word-graph based approach.

Table 5.2: For each off-line HTR task: statistics of the sets of on-line words used as feedback to correct the off-line HTR and baseline performance (classification error – ER) of the corresponding on-line HTR subsystem without using any interaction-derived contextual information (using plain 1-grams).

Task	#Words	#Uniq-Words	Lexicon	ER(%)
ODEC-M3	753	378	2 790	5.1
IAMDB	755	510	8 017	4.6
CS-page	1 196	648	2 277	6.4
CS-book	1 487	703	2 237	6.1

Note that these ER values are obtained without taking advantage of any interaction-derived contextual information (i.e., just using plain uni-grams). Therefore these figures represent the highest accuracies that could be expected if, e.g., an off-the-shelf on-line HTR system were adopted to implement the MM-CATTI feedback decoder.

Table 5.3: Writer average MM-CATTI feedback decoding error rates for the different corpora and three language models: plain unigram (U , *baseline*), error-conditioned unigram (U_e) and prefix-and-error conditioned bigram (B_e). The relative accuracy improvement for B_e with respect to U is shown in the last column.

Corpus	Feedback ER (%)			Rel. Improv. (%) B_e
	U	U_e	B_e	
ODEC-M3	5.1	5.0	3.1	39.2
IAMDB	4.6	4.3	3.5	23.9
CS-page	6.4	6.2	5.8	9.3
CS-book	6.1	5.9	5.5	8,2

As explained in Section 5.2 information derived from the interaction process can be taken into account in order to improve the accuracy of the on-line HTR subsystem. Table 5.3 presents the writer average feedback decoding error rates for the ODEC, IAMDB, CS-page and CS-book corpora and three language models which embody increasingly strong interaction derived constraints. The first one is the plain unigram estimation of $P(d)$, already reported in Table 5.2 as a *baseline*. The second is an error-conditioned unigram estimation of $P(d | e)$ (Equation (5.4)). The third model is a prefix-and-error conditioned bigram estimate of $P(d | \mathbf{p}', e)$ (Equation (5.5)). All these models are derived from the original language

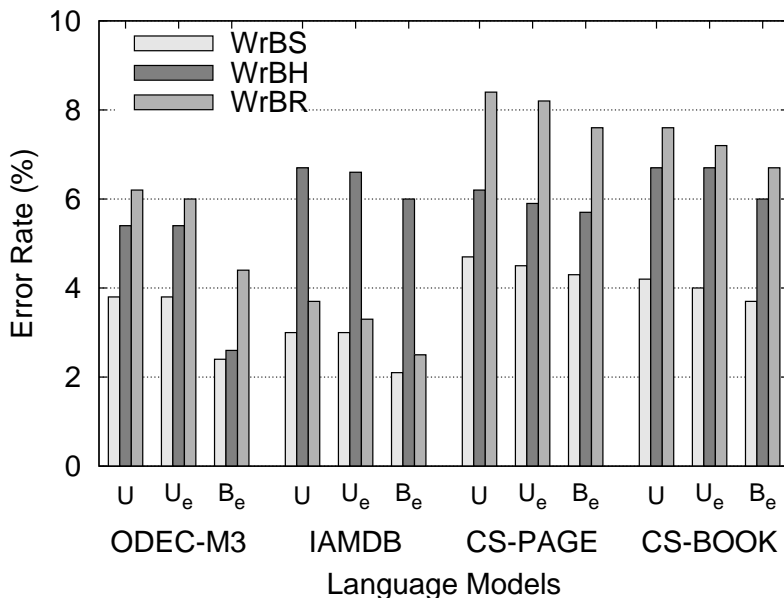


Figure 5.5: MM-CATTI feedback decoding error rates for different writers, corpora and prefix-constrained language models.

models employed for the main, off-line HTR system, as explained in Section 5.4. As observed in Table 5.3, feedback decoding accuracy increases significantly as more interaction-derived constraints are taken into account.

Individual recognition error rates of each of the three UNIPEN writers used in the experiments are plotted in Figure 5.5 for the different language models and corpora. The accuracy for the three writers is very similar, being WrBS who better results obtained.

Table 5.4 summarizes all the CATTI and MM-CATTI results obtained in this work. The third and fourth columns show the CATTI WSR decomposed into the percentage of corrections successfully made through the on-line HTR feedback modality (the touch-screen, TS) and those for which the feedback decoder failed and the correction had to be entered by means of the keyboard (KBD). These figures correspond to the three-writers averaged decoding errors reported in Table 5.3. The last two columns show the overall estimated effort reductions (EFR) in the CATTI and MM-CATTI approaches. The MM-CATTI EFR is calculated under the simplifying (but reasonable) assumption that the cost of keyboard-correcting a feedback on-line decoding error is similar to that of another on-line touchscreen interaction step. That is, each KBD correction is counted twice: one for the failed touch-screen attempt and another for the keyboard correction itself.

According to these results, the expected user effort for the more ergonomic and user-preferred touch-screen based MM-CATTI, is only slight higher than that of CATTI in the ODEC and on the IAMDB corpora. On the CS corpora the results shown that the expected user effort is very similar to the expected effort on a post-editing system. However, this extra

human effort entails an human-machine interaction more easier and comfortable.

Table 5.4: From left-to-right: post-editing corrections (WER), interactive corrections needed (WSR), contributions of both input modalities: on-line touch-screen (TS) and keyboard (KBD), and overall estimated effort reduction (EFR) achieved by the proposed approaches. All results are percentages.

Corpus	Post-edit WER	CATTI WSR	MM-CATTI		Overall EFR	
			TS	KBD	CATTI	MM-CATTI
ODEC	22.9	18.8	18.1	0.7	17.5	14.8
IAMDB	25.3	21.1	20,4	0,7	16.6	13.8
CS-page	28.5	26.9	25.4	1.5	5.6	0.4
CS-book	33.5	32.1	30,4	1,7	4.2	-0,8

5.7 Conclusions

In this chapter the use of on-line touch-screen handwritten pen strokes is studied as an alternative mean to input the required word CATTI corrections. We have called this new approximation “multimodal CATTI” (MM-CATTI). From the results, we observe that (in many cases) this much more ergonomic feedback modality can be implemented without significantly increasing the number of interaction steps due to errors caused by the decoding of the feedback signals. This is achieved thanks to the constraints derived from the interactive process.

It should be mentioned here that, in addition to the laboratory experiments reported in previous section, a complete MM-CATTI prototype has been implemented (see Chapter 6) and already submitted to preliminary, informal tests with real users. According to these tests, the system does meet the expectations derived from the laboratory experiments; both in terms of usability and performance. This is particularly true for the on-line HTR feedback decoding accuracy: even though the on-line HTR HMMs were trained from artificially built words using UNIPEN character samples, the accuracy in real operation with real users is observed to be similar to that shown in the laboratory results here reported. Of course even higher accuracy can be easily achieved by retraining the models with the text handwritten by the actual users.

Bibliography

- [TRPV09] Alejandro Toselli, Verónica Romero, Moisés Pastor, and Enrique Vidal. Multimodal interactive transcription of text images. *Pattern Recognition. In Press, Accepted Manuscript*, 2009. doi = 10.1016/j.patcog.2009.11.019.
- [TRV08] A. H. Toselli, V. Romero, and E. Vidal. Computer assisted transcription of text images and multimodal interaction. In *Proceedings of the 5th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 5237 of *Lecture Notes in Computer Science*, pages 296–308. Utrecht, The Netherlands, September 2008.

CHAPTER 6

A Web-based Demonstrator to Interactive Multimodal Transcription

6.1 Introduction

In this chapter a web-based demonstrator of the interactive multimodal approach presented in this thesis is introduced. In this web-based demonstrator the user feedback is provided by means of pen strokes on a touchscreen [RLA+09, RLTV09]. The user feedback allows to improve the system accuracy, while multimodality increases system ergonomics and user acceptability. Figure 1.1 (left) of the Chapter 1 shows a user interacting with this demo. Both the original image and the system's transcription hypotheses can be easily aligned and jointly displayed on the touchscreen (right).

Although, the actual system is only a demonstrator, in a next future, such web-based MM-CATTI system will allow to carry out collaborative tasks with thousands of user across the globe, thus reducing notably the overall image recognition process. Since the users operate within a web browser window, the system also provides cross-platform compatibility and require no disk space on the client machine.

It is important to say here, that this demonstrator has been developed with the collaboration of several persons. Specifically, the web interface has been developed by Luis A. Leiva and the Application Programming Interface (API) that allows client and server applications to communicate through sockets has been developed by Vicent Alabau.

A description of the user interaction protocol of the proposed demonstration is given in Section 6.2. Then, an outline about the demo is detailed on Section 6.3. Finally, some results and conclusions are drawn in Section 6.4.

6.2 User Interaction Protocol

In the MM-CATTI web-based demonstrator, the user is directly involved in the transcription process, where following a preset protocol, he/she is responsible of validating and/or correcting the HTR output during the process. The protocol that rules this process was presented in the Chapter 5 and can be summarized in the following steps:

- The HTR system proposes a full transcription of the input handwritten text image.
- The user validates the longest prefix of the transcription which is error-free and enters some on-line touchscreen pen-strokes and/or some amendment keystrokes to correct the first error in the suffix.
- If pen strokes are available, an on-line HTR feedback subsystem is used to decode this input.
- In this way, a new extended consolidated prefix is produced based on the previous validated prefix, the on-line decoding word and the keystroke amendments. Using this new prefix, the HTR suggests a suitable continuation of it.
- These previous steps are iterated until a final, perfect transcription is produced.

The interaction between the user and the system is not only limited to write the full correct word, but other different operations can be carried out using both pen-strokes and/or keystrokes. The types of operations that can be carried out are:

- Substitution: The first erroneous word is substituted by the correct word. The accepted prefix consists of all the words preceding the substituted word and the new correct word.
- Deletion: The incorrect word is deleted. The accepted prefix consists of all the words preceding the deleted word plus the word that follows the deleted word.
- Single click validation: This operation correspond with the Mouse Action operation studied in Section 4.5. All the words that precede the incorrect word constitute the validated prefix. The system proposes a new suffix where the first word is different to the incorrect word.
- Insertion: A new word is inserted. The validated prefix are all the word precedent the inserted word, the inserted word and the word that follows the inserted word.
- Accept Final Transcription: The proposed transcription is validated.

6.3 System description

The demonstrator presented in this chapter is publicly available at “<http://catti.iti.upv.es>”. On this web-based demo the client-server communication is made through sockets. Using sockets has several advantages. On the one hand the interaction process is quite faster and,

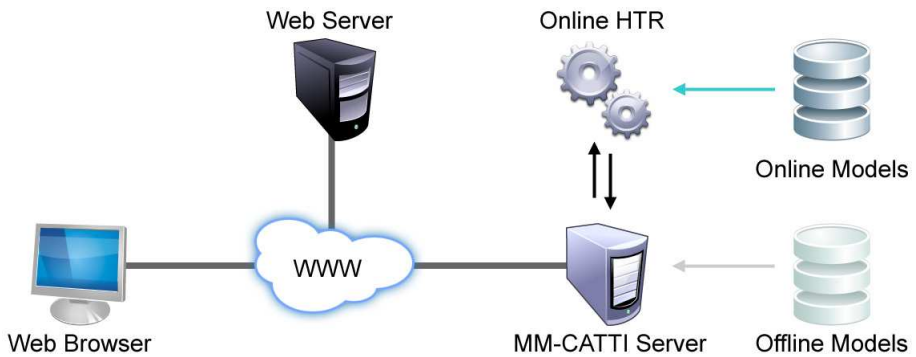


Figure 6.1: System architecture. First, the web client requests a web page with an index of all available pages in the document to be transcribed. The user then navigates to a page and begins to transcribe the handwritten text images line by line. She can make corrections with pen strokes and also use the keyboard. If pen strokes are available, the MM-CATTI server uses an on-line HTR feedback subsystem to decode them. Then, taking into account the decoded word and the off-line models, the MM-CATTI server responds with a suitable continuation to the prefix validated by the user. All corrections are stored in plain text logs on the MM-CATTI server, so the user can retake them in any moment.

on the other hand, the multiuser environment can be easily implemented, so that, several users across the globe can work concurrently on the same task. In addition, the web server and the MM-CATTI server do not need to be physically at the same place. Therefore, a dedicated transcription server can be run per task to deal with high CPU demanding corpora, or several servers can be set up with the same task to serve an increasing amount of users. Figure 6.1 shows a schematic view of the system's architecture.

On the next subsections the API, the MM-CATTI server and the Web Interface are described.

6.3.1 Application Programming Interface

Based on the previously presented protocol, a generic subset of primitives were extracted, and a client-server Application Programming Interface (API) that allows client and server applications to communicate through sockets was designed.

Three basic functions summarize the API:

- **set_source** : selects the source phrase to be transcribed.
- **set_prefix** : sets the longest error free prefix and amends the first error with the keyboard.
- **set_prefix_online** : sets the longest error free prefix and amends the first error with pen strokes.

6.3.2 MM-CATTI server

The MM-CATTI server combines all the information received from the client and compute a suitable solution. It follows the approach presented on Chapter 5, where both online and offline HTR systems are based on HMM and n -gram language models.

The offline system is implemented using word-graphs. These word-graphs are a pruned version of the Viterbi search trellis obtained when transcribing the whole image sentence. In order to make the system able to interact with the user in a time efficient way, they are computed beforehand.

Once the user selects the line to be transcribed, the client application send to the MM-CATTI server the `set_source` message. The MM-CATTI server loads the word-graph corresponding to the selected line and proposes a full transcription as explained in the Section 3.2 of the Chapter 3.

When the user makes some correction, if pen strokes are available, the MM-CATTI server uses an on-line HTR feedback subsystem to decode them. After preprocessing and extracting the features, as it is explained in Section 3.3 of the Chapter 3, the pen strokes are decoded following the last scenario presented in Chapter 5, taking into account information derived from the validated prefix and the previous suffix as shown Equation (5.5).

Once the pen strokes have been decoded, a new prefix can be generated taking into account the validated prefix, the new decoded word and the operation that the user has carried out (substitution, deletion, insertion, ...). Then, this new prefix is parsed on the off-line word-graph and a suitable continuation of it is provided following techniques described in Chapter 4. It can be possible that the prefix is not on the word-graphs, so, the error correcting parsing explained in Section 4.4.2 is applied.

All the corrections made by the user are stored in plain text logs on the MM-CATTI server. In this way, the user can retake them in any moment.

6.3.3 Web Interface

The Web Interface is responsible for showing the user interface and capturing the user actions on the different modalities of interaction, i.e, keyboard and pen strokes.

On the main page of the demonstrator ("<http://catti.iti.upv.es>") does not only appear the link to the demonstrator, but a lot of information related with it, such as videos, projects, awards, publications or teaching (Figure 6.2 shows a screen capture of this main page). To begin to work with the demo, the user must select the option "*Try a live MM-CATTI demo*" and a new page with the different documents to transcribe will appear (see Figure 6.3). The user must choose one of the available documents by clicking on the "*transcribe*" button. By clicking on "*use custom server?*" link, the user can specify a custom CATTI server while her session is active (see Figure 6.4).

Once the user has selected the document to transcribe, a new web page with an index of all pages in the corpus appears, allowing the user navigate to any page. On Figure 6.5 we can see the different pages of the legacy handwriting document from the nineteenth century "*Cristo Salvador*".

To begin with, once the user selects a thumbnail page from the index, the full page is loaded (Figure 6.6). The center block is the page to transcribe itself. The tidy menu on the

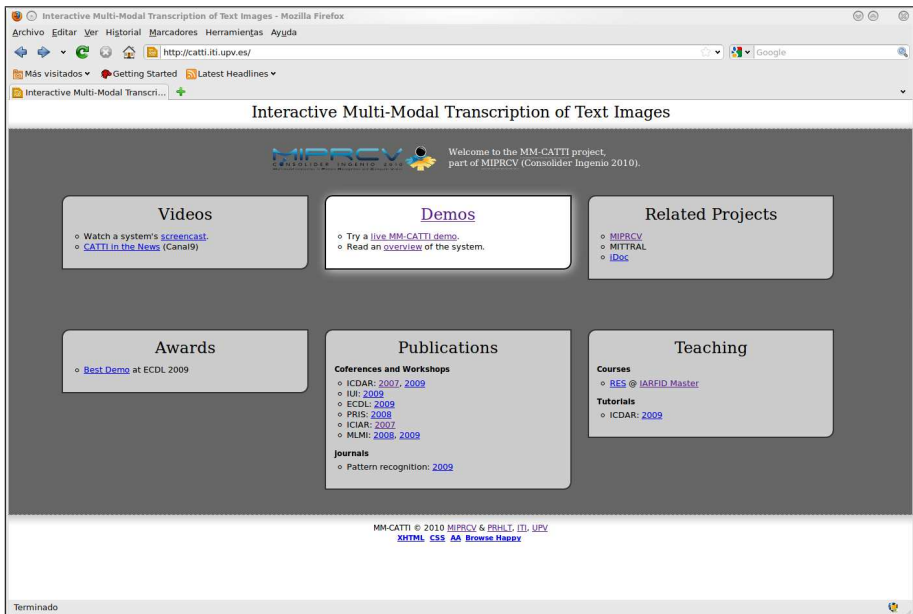


Figure 6.2: The main page of the web-based demonstrator. In addition the link to the demonstrator, a lot of information related with it is shown: videos, projects, awards, publications and teaching.

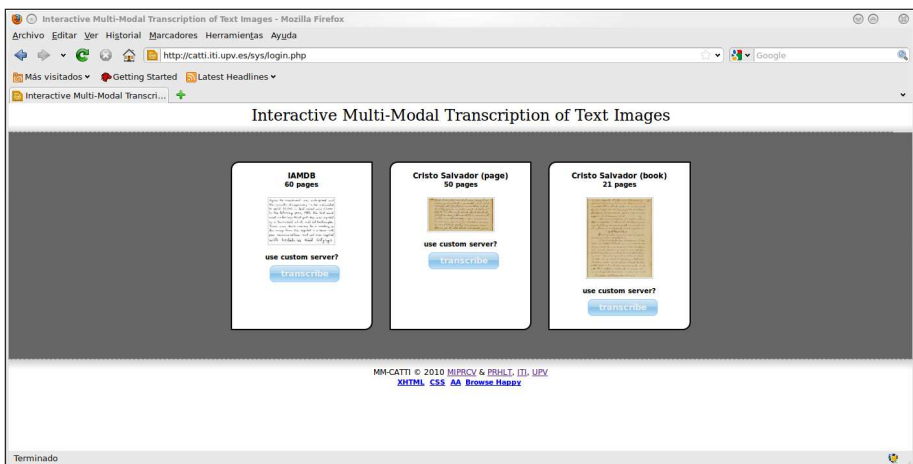


Figure 6.3: All the documents to transcribe.

right side is a pagination item. With that pagination item the user can browse all the pages quickly. By using the page browser located above the center page, the user can browse all



Figure 6.4: By clicking on “*use custom server?*” link, the user can specify a custom CATTI server while her session is active.

the pages visually. The bottom menu is intended to help the user with common tasks, such as closing session, changing document, displaying application shortcuts, or exploring the API.

Then, the user can select a line from the current page by clicking on its image, and the system will propose an initial, full transcription (see Figure 6.7). If no error is detected, the user chooses another text line image to be transcribed. Otherwise, the user validates the longest prefix of the transcription which is error-free and corrects the first error in the suffix.

If an *e-pen* is available, the system uses an on-line HTR feedback subsystem to recognize the user corrective pen-strokes. Then, taking into account the (multimodal) user corrections, the system responds with a suitable continuation to the prefix validated by the user.

The interaction between user and system can be carried out using pen-strokes on a touch-screen or using traditional peripherals like keyboard and mouse. On the next subsections these different interactions modes are explained.

6.3.4 Electronic Pen or Touchscreen Interaction

This is the default interaction mode. The application can be used with any kind of pointing device, such as a electronic pen or a PC tablet. The computer mouse can be also used. However, this option is discouraged. That is because the computer mouse, unfortunately, is not too good at writing with precision.

By using pen-strokes, the user can directly write the correct word, or also introduce some gestures to indicate different error types. These intuitive pen-based gestures greatly simplify the error correction effort. On Figure 6.8 the different gestures to interact with the system are shown:

- Substitution: Place the e-pen over an incorrect word and write down the correct text.
- Deletion: Drawing a diagonal line over an incorrect word and it will be deleted. The deletion gesture must begin outside the text field boundaries.

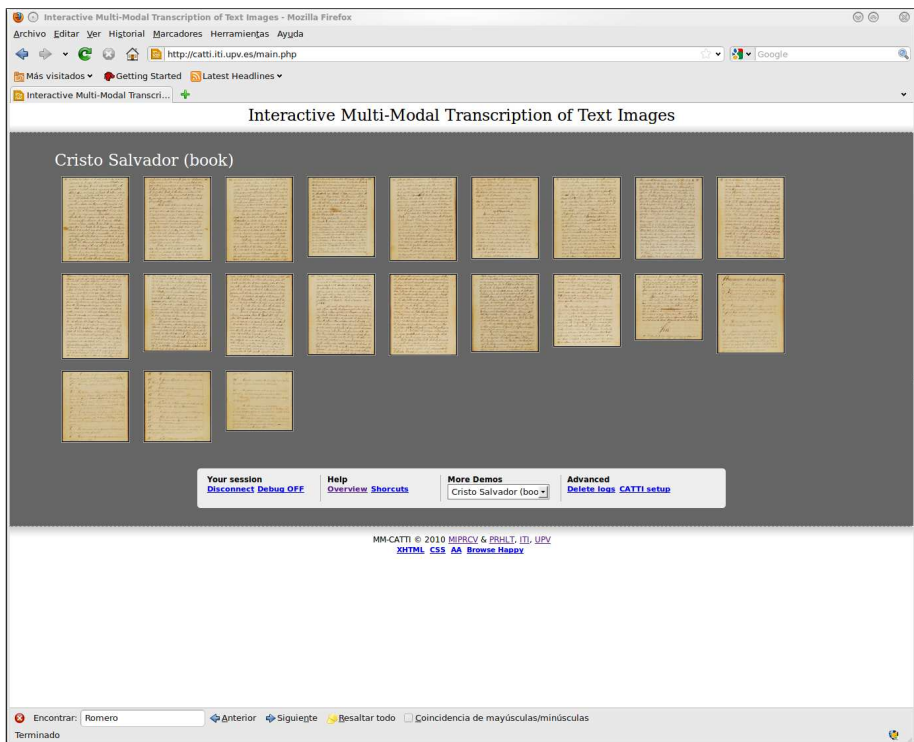


Figure 6.5: A thumbnail for each page of the document chosen by the user is shown. In this case the thumbnails belong to pages of the “Cristo Salvador” book.

- **Single Click Validation:** By clicking on a word the system proposes a new suffix where the clicked word has changed. This gesture can be understood as a “Rejection” operation.
- **Insertion:** Draw a vertical line between two words and then write down the text to insert.
- **Accept Final Transcription:** Draw a verification gesture (v-like) after the last corrected word to accept the full proposed transcription. The next image line will be loaded automatically.

6.3.5 Keyboard Interaction

If the user prefers, she can work with traditional peripherals like keyboard and mouse. To switch to keyboard mode any key over the application interface must be pressed. It is important to remark that while the stylus pen is moved over the words the focus is updated. So, since the keys work as expected, take these examples when switching to keyboard mode:

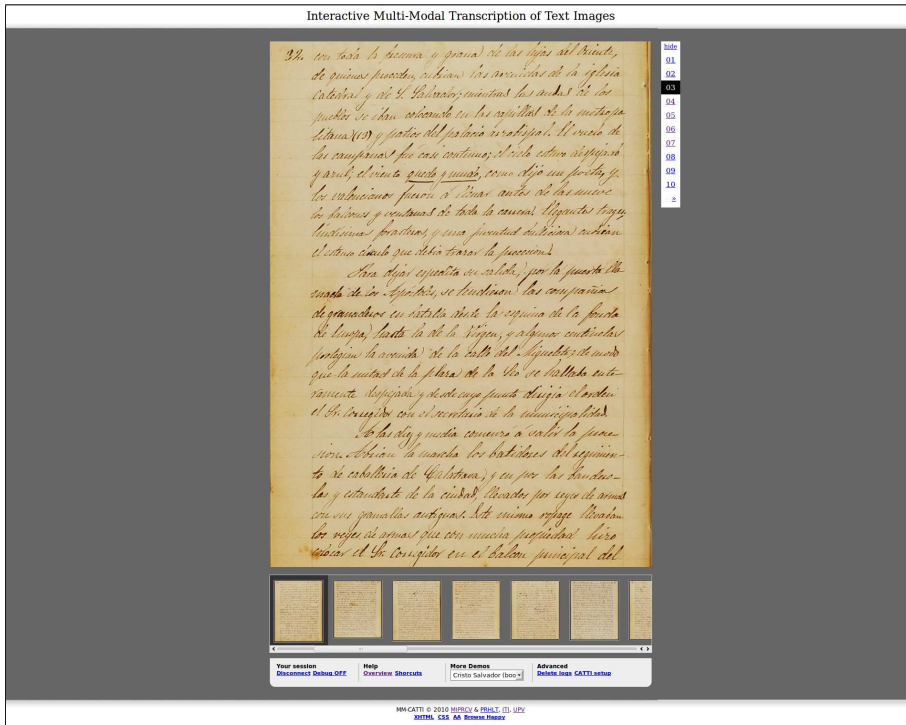


Figure 6.6: The selected page to be transcribed.

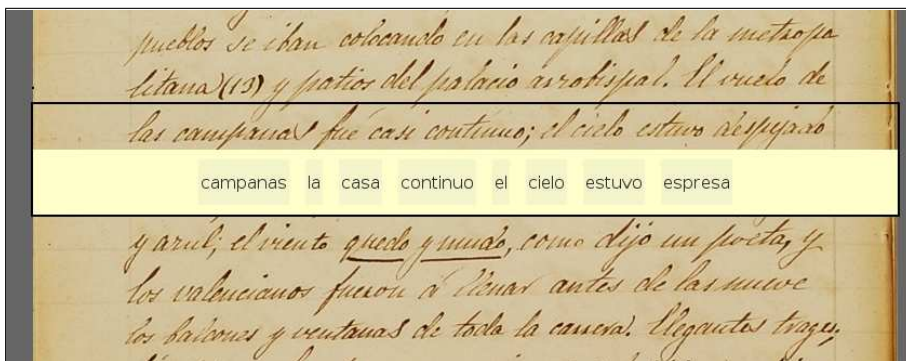


Figure 6.7: The HTR system proposes a full transcription of the input handwritten text line image.

- Clicking the space or delete key will delete the text.
- Clicking the TAB key will change to the next word in the tab loop.

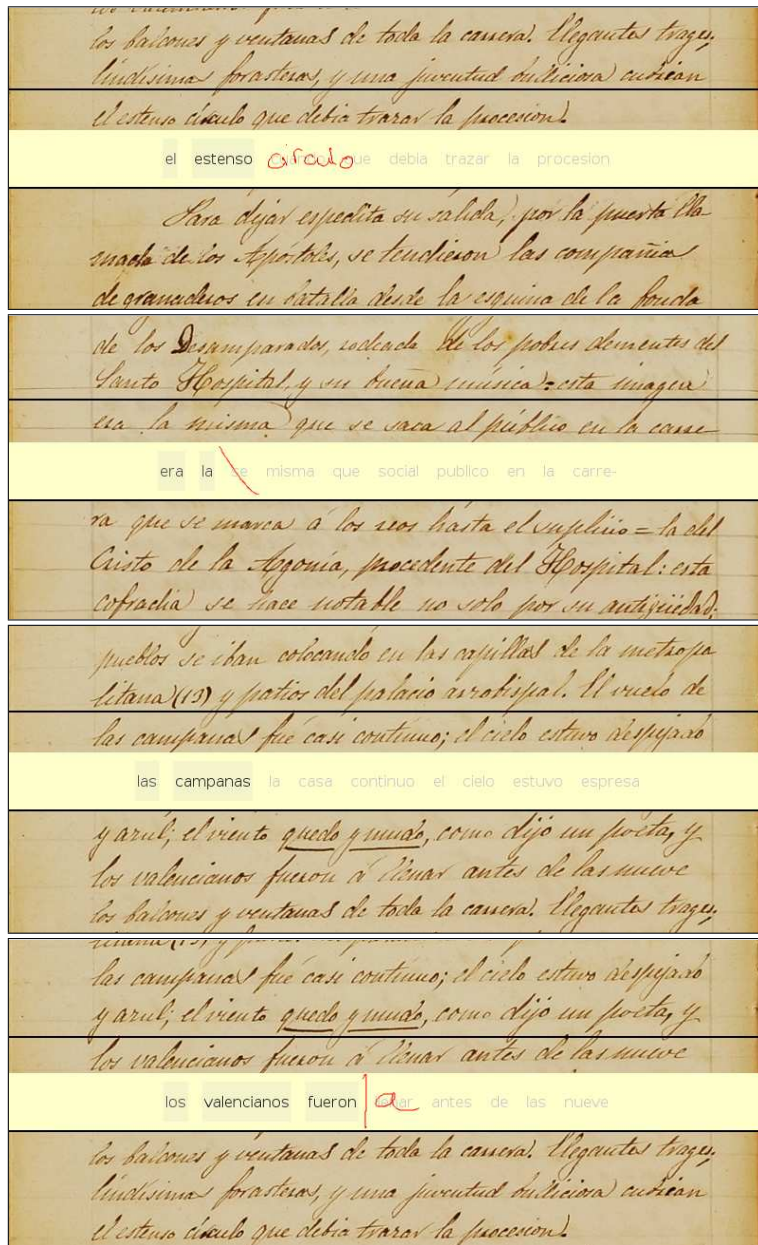


Figure 6.8: Four interaction gestures to generate and/or validate an error-free prefix. From top to bottom: substitution, deletion, single click validation and insertion.

- Clicking any alphanumeric or punctuation key will overwrite the word text.
- Clicking non-character keys (UP, CTRL, CAPS, ALT, etc.) will give focus to the current word.

On Figure 6.9 is shown an example of the demo in the keyboard interaction mode. The different interactions in the keyboard mode are:



Figure 6.9: The keyboard mode.

- Substitution: Once the incorrect word has been changed, pressing ENTER or going to the next word (with the TAB key) the validated prefix will be sent.
- Deletion: Pressing “CTRL + DEL” will delete the selected text field.
- Single click validation: this is similar to the “Single click validation” presented on the previous subsection, but now using the mouse instead the e-pen. This also can be carried out using the keyboard, pressing “CTRL + UP” the system will propose a new transcription and the previous one will be stored on a internal buffer. Pressing “CTRL + DOWN” will load the previous transcription from the internal buffer.
- Insertion: Pressing the keys “CTRL + SPACE” will insert a new text field to the right of the selected word. If the “SHIFT” key is also pressed, the text field will be inserted to the left of the selected word instead.
- Accept Final Transcription: Pressing “CTRL + ENTER” the full proposed transcription will be accepted. The final transcription will be until the text field with focus. The next image line will be loaded automatically.

6.4 Results and Conclusions

The results of the automatic evaluation metrics discussed on the previous chapter were intended to give us a rough idea of how the system could be expected to perform when used by

real transcriber. In addition to the laboratory experiments, the complete MM-CATTI prototype presented in this chapter allow us to carry out informal test with real users.

According to these tests, the system does meet the expectations derived from the laboratory experiments. This is particularly true for the on-line HTR feedback decoding accuracy: even though the on-line HTR HMMs were trained from artificially built words using UNIPEN character samples. The accuracy in real operations with real users is observed to be similar to that shown in the laboratory results. From the tests also follows that the interactive scenario implemented in the MM-CATTI server is really more comfortable and friendlier to the user than the post-editing approach.

The obvious next step is to carry out formal field tests to assess the validity of our assumptions and estimations under real working conditions of expert (paleography) transcribers of handwritten documents.

Bibliography

- [RLA+09] V. Romero, L. Leiva, V. Alabau, A. Toselli, and E. Vidal. A web-based demo to interactive multimodal transcription of historic text images. In *Proceedings of the 13th european conference on digital libraries (ECDL)*, volume 5714 of *LNCS*, pages 459–460. Corfu, Greece, September 2009.
- [RLTV09] V. Romero, L. A. Levia, A. H. Toselli, and E. Vidal. Interactive multimodal transcription of text image using a web-based demo system. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 477–478. Sanibel Island, Florida, February 2009.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

In this work, we have proposed a new interactive, on-line approach to the transcription of handwritten documents, which combines the efficiency of automatic HTR systems with the accuracy of users (e.g. paleography experts). This approach is based on a recently introduced framework called ‘interactive predictive’ (IP) processing (IPP) [VRCGV07]. We have called this approach “Computer Assisted Transcription of Text Images” (CATTI). Here, the corrections made by a human transcriber become part of a prefix of the final target transcription. This prefix is used by CATTI to suggest a new suffix that the human transcriber can accept or modify in an iterative way until a satisfactory, correct target transcription is finally produced. Empirical tests presented in this work clearly support the benefits of using this approach rather than traditional HTR followed by human post-editing. Two different CATTI implementations have been studied. The first of them consists in building a special language model and the second one in more sophisticated word-graph techniques. In the word-graph based implementation, it was necessary to adapt, implement and integrate efficient error-correcting parsing algorithms in order to guarantee low response time while preserving adequate transcriptions.

The use of Mouse Actions (MA) as an additional information source has been also studied. As soon as the user points out the place where the next error is found, the system proposes a new, hopefully more correct, continuation, thereby trying to anticipate up-coming user corrections. Obtained results show that a significant benefit can be obtained, in terms of word-stroke reductions. It is worth noting that alternative (n-best) suffixes could also be obtained with the conventional CATTI system. However, by considering the rejected words

to propose the alternative suffixes, the MA interaction method is more effective and more comfortable for the user.

The interaction in the CATTI system presented comes in the form of whole-word interactions. However, in order to make the interaction process easier and more comfortable to the user, a character-level interaction has been studied. Considering the results obtained in the experiments, we can conclude that using this interaction level not only allows for more ergonomic and friendly interfaces, but significant amounts of human effort in the handwritten text transcription process can be saved. It is worth noting, however, that results obtained at the character and word levels are not directly comparable. A word-level correction encapsulates fairly well all the cognitive and physical human efforts needed to locate an error and type the correction. This is true both for off-line editing Word Error Rate (WER) and for CATTI corrections Word Stroke Ratio (WSR) and therefore word-level Estimated Effort Reduction (EFR) figures can be considered quite fair. In contrast, character level corrections are preferred by users, but it is unclear whether only the number of keystrokes can be fairly used for assessment purposes. A corrective keystroke generally needs no significant cognitive effort since, in most cases, it is part of the correction of an already detected word error. In other words, nor the Character Error Rate (CER) neither the Key Stroke Ratio (KSR) account well for the cognitive component of corrective actions and it does not seem easy to establish a single, adequate scalar score that captures correctly the two kinds of human efforts involved at the character level.

We have also studied the use of on-line touch-screen handwritten pen strokes as a complementary means to input the required CATTI correction feedback. We call this multimodal approach “MM-CATTI”. From the results, we observe that the use of this more ergonomic feedback modality comes at the cost of only a reasonably small number of additional interaction steps needed to correct the few feedback decoding errors. The number of these extra steps is kept very small thanks to the MM-CATTI ability to use interaction-derived constraints to considerably improve the on-line HTR feedback decoding accuracy. Clearly, this would have not been possible if just a conventional, off-the-shelf on-line HTR decoder were trivially used for the corrections steps.

The advantage of CATTI and MM-CATTI over traditional HTR followed by post-editing goes beyond the good estimates of human effort reductions achieved. When difficult transcription tasks with high WER are considered, expert users generally refuse to post-edit conventional HTR output. In contrast, the proposed interactive approaches constitute a much more natural way of producing correct text. With an adequate user interface, CATTI or MM-CATTI let the users be dynamically in command: If predictions are not good enough, then the user simply keeps typing at his/her own pace; otherwise, he/she can accept (partial) predictions and thereby save both thinking and typing effort.

We should mention here that, in addition to the laboratory experiments reported in previous chapters, a complete MM-CATTI prototype has been implemented and already submitted to preliminary, informal tests with real users. According to these tests, the system does meet the expectations derived from the laboratory experiments. This is particularly true for the on-line HTR feedback decoding accuracy: even though the on-line HTR HMMs were trained from artificially built words using UNIPEN character samples, the accuracy in real operation with real users is observed to be similar to that shown in the laboratory results here reported. Of course even higher accuracy can be easily achieved by retraining the on-line models with

the text handwritten by the actual users.

Summarizing the main contributions of this thesis are the following:

1. First a set of experiments have been carried out to tune the different parameters of the off- and on-line HTR systems. In this way results comparable with the state-of-the-art have been obtained for the different tasks used in the experiments.
2. The interactive predictive framework has been developed and successfully tested in HTR. A computer assisted transcription of handwritten text images (CATTI) system has been developed based on HMMs and n -gram language models. The system has been automatically tested on three corpora of handwritten text images and, the results suggest that, using the interactive approach, considerable amounts of user effort can be saved with respect to post-editing the output of non-interactive HTR systems.
3. Mouse Actions (MA) as an addition information source have been studied. The CATTI system can automatically change its prediction as soon as the user points out the place where the next error is found. This way, many explicit user corrections are avoided.
4. Character-level keystroke interaction has been studied and new empirical measures to compare the estimated user-effort reduction obtained by CATTI with respect to conventional post-editing at character level has been introduced. The system has been fully developed and then tested on three corpora. Considering the results, we can conclude that using this interaction level, a significant amount of human effort can be saved and, moreover, it allows for more ergonomic and friendly interfaces.
5. A direct adaptation of the Viterbi algorithm to implement the CATTI system would lead to a computational cost that grows quadratically with the number of words of each sentence. This can be problematic for large sentences and/or for fine-grained (character-level) interaction schemes. Here word-graph techniques that can achieve very efficient, linear cost search have been used. In addition, we adapt well-known error-correcting algorithms in order to be integrated into the word-graph based CATTI system. This system has been evaluated on three handwritten tasks with good results.
6. A touchscreen interface has been studied in order to obtain an easier and more comfortable human-machine interaction. The touchscreen interaction is perhaps the most natural modality to provide the required feedback on the CATTI system. The on-line feedback HTR subsystem is based on HMMs and n -gram language models. The resulting multimodal system, called MM-CATTI, has been tested using an on-line handwritten corpus and three off-line handwritten corpora, with promising results.

7.2 Publications related with this work

Parts of this thesis has been published in international workshops, conferences and journals. In this section, we review these publications pointing out their relation with this thesis.

- Publications related with new technologies for conventional (non-interactive) handwritten text recognition (Chapter 1):

- **Verónica Romero**, Adrià Giménez, and Alfons Juan. “Explicit Modelling of Invariances in Bernoulli Mixtures for Binary Images”. In *3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2007)*, volume 4477 of Lecture Notes in Computer Science (LNCS), pages 539-546. Springer-Verlag, Girona (Spain), 2007.
- Publications related with (conventional, HMM-based) technology used in handwritten text recognition and the preprocessing techniques (Chapter 3):
 - Moisés Pastor, Alejandro H. Toselli, **Verónica Romero**, and Enrique Vidal. “Improving handwritten off-line text slant correction”. In *Proceedings of the Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, pages 389-394. Palma de Mallorca, Spain, 2006.
 - **Verónica Romero**, Moisés Pastor, Alejandro H. Toselli, and Enrique Vidal. “Criteria for handwritten off-line text size normalization”. In *Proceedings of the Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, pages 395-399. Palma de Mallorca, Spain, 2006.
 - Alejandro H. Toselli, Moisés Pastor, **Verónica Romero**, Alfons Juan, Enrique Vidal, and Francisco Casacuberta. “Off-line and On-line Continuous Handwritten Text Recognition in PRHLT Group”. Centre de Visió per Computador (UAB). Pattern Recognition: Progress Directions and Applications (ISBN 84-933652-6-2), volume 10, pages 146-161, 2006.
 - **Verónica Romero**, Vicent Alabau, and Jose Miguel Benedí. “Combination of N-grams and Stochastic Context-Free Grammars in an Offline Handwritten Recognition System”. In *3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2007)*, volume 4477 of LNCS, pages 467-474. Springer-Verlag, Girona (Spain), 2007.
 - Alejandro H. Toselli, **Verónica Romero**, and Enrique Vidal. “Viterbi Based alignment between Text Images and their Transcripts”. In *Language Technology for Cultural Heritage Data (LaTeCH 2007)*, pages 9-16. Prague, Czech Republic, 2007.
 - **Verónica Romero**, Alejandro H. Toselli, and Enrique Vidal. “Aligning handwritten text images and transcriptions of historic documents”. In *Proceedings of the 2nd EVA 2008 Vienna Conference*, pages 87-94. Vienna, Austria, 2008.
- Publications related with the technology used on the Computer Assisted Transcription of Handwritten Text Images (CATTI) system presented on Chapter 4:
 - Alejandro H. Toselli, **Verónica Romero**, Luis Rodríguez, and Enrique Vidal. “Computer Assisted Transcription of Handwritten Text”. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 944-948. IEEE Computer Society, Curitiba, Paraná, Brazil, 2007.
 - **Verónica Romero**, Alejandro H. Toselli, Luis Rodríguez, and Enrique Vidal. “Computer Assisted Transcription for Ancient Text Images”. In *International Conference on Image Analysis and Recognition (ICIAR 2007)*, volume 4633 of LNCS, pages 1182-1193. Springer-Verlag, Montreal (Canada), 2007.

-
- Antonio L. Lagarda, Vicent Alabau, Carlos Martínez-Hinarejos, Alejandro H. Toselli, **Verónica Romero**, Jose Ramon Navarro, and Enrique Vidal. “Computer-assisted handwritten text transcription using speech recognition”. In *V Jornadas en Tecnología del Habla (VJTH’2008)*, pages 229-232. Bilbao, Spain, 2008.
 - **Verónica Romero**, Alejandro H. Toselli, Jorge Civera, and Enrique Vidal. “Improvements in the computer assisted transcription system of handwritten text images”. In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems (PRIS 2008)*, pages 103-112. Barcelona, Spain, 2008.
 - **Verónica Romero**, Alejandro H. Toselli and Enrique Vidal. “Using Mouse Feedback in Computer Assisted Transcription of handwritten Text Images”. In *International Conference on Document Analysis and Recognition (ICDAR 2009)*, pages 96–100. Barcelona, Spain, 2009.
 - **Verónica Romero**, Alejandro H. Toselli and Enrique Vidal. “Character-level interaction in Computer-Assisted Transcription of Text Images”. In *International Conference on Frontiers in Handwritten Recognition (ICFHR 2010)*. Kolkata, India, 2010. To be published.
 - **Verónica Romero**, Alejandro H. Toselli and Enrique Vidal. “Computer assisted transcription of text Images: Results on The GERMANA corpus and analysis of improvements needed for practical use”. In *International Conference on Pattern Recognition (ICPR 2010)*. Istanbul, Turkey, 2010. To be published.
- Publications related with the Multimodal version of the CATTI system (Chapter 5):
 - Alejandro H. Toselli, **Verónica Romero** and Enrique Vidal. “Computer Assisted Transcription of Text Images and Multimodal Interaction”. In *Proceedings of the 5th Joint Workshop on Machine Learning and Multimodal Interaction (MLMI 2008)*, volume 5237 of LNCS, pages 296-308. Utrecht, The Netherlands, 2008.
 - Alejandro H. Toselli, **Verónica Romero**, Moisés Pastor and Enrique Vidal. “Multimodal Interactive Transcription of Text Images”. In *Pattern Recognition*. 43, 5, pages 1814-1825, 10.1016/j.patcog.2009.11.019, A.
 - Oriol Ramos Terrades, Alejandro H. Toselli, Nicolás Serrano, **Verónica Romero**, Enrique Vidal and Alfons Juan. “Interactive layout analysis and transcription systems for historic handwritten documents”. In *10th ACM Symposium on Document Engineering (DocEng2010)*. Manchester UK, 21-24 September 2010. To be published.
 - Alfons Juan, **Verónica Romero**, Joan Andreu Sánchez, Nicolás Serrano, Alejandro H. Toselli, Enrique Vidal. “Handwritten Text Recognition for Ancient Documents”. In *Workshop on Applications of Pattern Analysis (WAPA 2010)*. Cumberland Lodge, 1-2 September 2010. To be published.
 - Publications related with the MM-CATTI prototype presented in Chapter 6:
 - **Verónica Romero**, Luis A. Leiva, Vicent Alabau, Alejandro H. Toselli and Enrique Vidal. “A Web-Based Demo to Interactive Multimodal Transcription of

Historic Text Images”. In *European Conference on Digital Libraries (ECDL 2009)*, Volume 5714 of LNCS, pages 459-460. Springer-Verlag, Corfu, Greece, 2009.

- **Verónica Romero**, Luis A. Leiva, Alejandro H. Toselli and Enrique Vidal: “Interactive Multimodal Transcription of Text Images Using a Web-based Demo System”. In *International Conference on Intelligent User Interfaces (IUI 2009)*, pages 477-478. Sanibel Island, Florida, 2009.
- Vicent Alabau, Daniel Ortiz, **Verónica Romero** and Jorge Ocampo: “A multimodal predictive-interactive application for computer assisted transcription and translation”. In *International Conference on Multimodal Interfaces (ICMI-MLMI 2009)*, pages 227-228. Cambridge, MA, USA, 2009.

7.3 Future work

A potential problem with the use of MM-CATTI in practice can arise when transcribing documents for which an adequate full lexicon cannot be established beforehand (often referred to as open-vocabulary operation). Following the tradition in Automatic Speech Recognition (ASR), in order to facilitate comparisons and reproducibility, all the MM-CATTI experiments here reported have been carried out under the closed-vocabulary assumption. However, practical transcription tasks typically entail open-vocabulary operation and some solution to this problem is needed. Of course, the lexicon of a task need not be strictly limited to the word-forms found in the (training) data available of this task. In practice, dictionaries or texts from other similar tasks or documents are often used to expand the word-forms found in the training data of the task considered. But some amount of residual out-of-vocabulary (OOV) word-forms must be expected. Furthermore, this can be really important in the case of ancient documents (one of the main MM-CATTI targets) where, for example, OOV words appear because of the frequent use of abbreviations/acronyms, which change from one era to another and even from one document to another [RTV10]. Thanks to the interactive nature of MM-CATTI operation, a simple way to cope with this problem is to progressively enrich the given lexicon by successively incorporating all the OOV tokens which appear in the document being processed [SJ10]. This way, a user correction will only be needed the first time a new word or abbreviation appears; later appearances will hopefully be correctly recognized by the system.

This can be seen as one of the simplest forms of adaptive learning, which is one of the main future research topics we plan to explore in the context of MM-CATTI. The correct transcriptions that are being continuously produced during the interactive work, can be advantageously used to dynamically and adaptively improve the underlying language and morphological off-line HTR models of the task or document(s) considered. In speech recognition, well known Adaptive Learning techniques exist for adapting the acoustic HMM models to the speaker and/or the audio channel (see for example [LW95, Woo01, LWK02, PN05]). These techniques are currently in use in many state-of-the-art recognition systems. In addition, Language Model adaptation is also possible ([KM90, Bel04, LNF04, SSJ10, SJ10]). In traditional ASR systems, these techniques require the user to provide adequate amounts of adaptation

data, which is not always possible, convenient or cost-effective. Typically, only small quantities of adaptation data can be used, which results in the systems not being able to take full advantage of adaptation techniques. In CATTI system, perfectly supervised adaptation data will be produced continuously, without the user even being aware of their production.

Similarly, the MM-CATTI feedback decoding accuracy can be easily (and significantly) improved by adaptively retraining the on-line HTR morphologic HMMs with the real pen-strokes which are being continuously produced by the specific user who is interacting with the system.

In addition to these Adaptive Learning ideas, our current and future works also aim at more directly taking further advantage of other interaction-derived informations. For instance, as mentioned in Chapter 5, feedback decoding accuracy can be further improved by using other informations derived from the main off-line HTR process. In particular, using the original text image being processed offers hope for significant improvements.

As explained in Chapter 4, the interaction between users and the CATTI system is not necessarily restricted to the recognition module. As a future work, a more general approach where the user could also interact with the preprocessing module in order to correct segmentation or preprocessing errors will be studied. The system could then use these corrections to improve the recognition accuracy. Following these ideas, there are some recent works [RTSG+10, ORTJ10] addressing interactive-predictive approaches applied to layout analysis. In this case, new assessment measures that take into account all the number of actions needed by the user in order to start from a digitized document and produce correct transcriptions will be defined.

Finally, our plans for future work include to carry out formal field tests to assess the validity of our assumptions and estimations under real working conditions of expert (paleography) transcribers of handwritten (historic) documents. Similar experiments to those conducted in [CCC+09] on a Computer Assisted Translation (CAT) system can be defined, where the results show that this suffix-predictive Interactive Machine Translation system can allow translators to increase their productivity while maintaining high-quality. However, as explained on Chapter 1, these experiments can be prohibitively expensive, because it will be necessary an entire panel of transcribers, to implement new professional user interfaces and to carry out a lot of experiments on a long period of time, that must be carefully analyse to avoid misleading conclusions. So, on a first approach experiments using the demonstrator presented on Chapter 6 and persons of our labs will be carried out.

Bibliography

- [Bel04] J.R. Bellegarda. Statistical language model adaptation: Review and perspectives. *Speech communication*, page 93, 2004.
- [CCC+09] F. Casacuberta, J. Civera, E. Cubel, A.L. Lagarda, G. Lapalme, E. Macklovitch, and E. Vidal. Human interaction for high quality machine translation. *Communications of the ACM*, 52(10):135–138, 2009.
- [KM90] R. Kuhn and R. De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and machine Intelligence*, page 570, 1990.
- [LNF04] G. Lapalme L. Nepveu, P. Langlais and G. H. Foster. Adaptive language and translation models for interactive machine translation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 190–197. Barcelona, Spain, July 2004.
- [LW95] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, page 171, 1995.
- [LWK02] H. Ney L. Welling and S. Kanthak. Speaker adaptive modeling by vocal trunc normalization. *IEEE Transactions on Speech and Audio Processing*, page 415, 2002.
- [ORTJ10] N. Serrano V. Romero E. Vidal O. Ramos Terrades, A.H. Toselli and A. Juan. Interactive layout analysis and transcription systems for historic handwritten documents. In *10th ACM Symposium on Document Engineering (DocEng2010)*, 2010.
- [PN05] M. Pitz and H. Ney. Vocal tract normalization equals linear transformation in cepstral space. *IEEE Transactions on Speech and Audio Processing*, page 930, 2005.
- [RTSG+10] O. Ramos Terrades, N. Serrano, A. Gordó, E. Valveny, and A. Juan. Interactive-predictive detection of handwritten text blocks. In *Document Recognition and Retrieval XVII*, volume 7534, 2010.
- [RTV10] V. Romero, A. H. Toselli, and E. Vidal. Computer assisted transcription of text images: Results on the germana corpus and analysis of improvements needed for practical use. In *International Conference on Pattern Recognition (ICPR 2010)*, Istanbul, Turkey, 2010. To be published.
- [SJ10] N. Serrano and A. Juan. The RODRIGO database. In *Proceedings of the The seventh international conference on Language Resources and Evaluation (LREC 2010)*, pages 19–21, Malta, May 2010.

- [SSJ10] N. Serrano, A. Sanchis, and A. Juan. Balancing error and supervision effort in interactive-predictive handwritten text recognition. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI 2010)*, pages 373–376, Hong Kong (China), February 2010.
- [VRCGV07] E. Vidal, L. Rodríguez, F. Casacuberta, and I. García-Varea. Interactive pattern recognition. In *Proceedings of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 4892 of *Lecture Notes in Computer Science*, pages 60–71. Brno, Czech Republic, June 2007.
- [Woo01] P. C. Woodland. Speaker adaptation for continuous density hmms: A review. In *Proceedings of the ICSA workshop on adaptation methods for Speech Recognition*. Sophia, Antipolis, France, 2001.

APPENDIX \mathcal{A}

Symbols and Acronyms

A.1 Symbols

$\Pr(\dots)$	the unknown “true” probability.
$P(\dots)$	the model probability.
a_{ij}	estate transition probability on an HMM from the state q_i to the state q_j : $a_{ij} \equiv a(q_i, q_j)$.
$b_i(x)$	emission vector x probability distribution function on an HMM for the state q_i : $b_i(x) \equiv b(q_i, x)$.
G	number of Gaussians on a mixture.
g	index for the Gaussians on a mixture.
μ_{jg}	mean vector for the component g in the Gaussian mixture of the state q_j on a HMM.
Σ_{jg}	covariance matrix for the component g in the Gaussian mixture of the state q_j on a HMM.
c_{jg}	weighting coefficient for the component g in the Gaussian mixture of the state q_j on a HMM.
$\mathbf{x} = x_1, x_2, \dots$	a sequence of feature vectors.
x_i and \vec{x}_i	the feature vector i of the sequence \mathbf{x} .
M	number of feature vectors on the sequence \mathbf{x} .
\mathbf{x}_i^j	subsequence of feature vectors extracted of the sequence \mathbf{x} , formed by the frames between the position i and j inclusive.
$\mathbf{w} = w_1, w_2, \dots$	sequence of words.
w_i	i th word in the word sequence \mathbf{w} .

l	number of words in the sequence \mathbf{w} .
\mathbf{w}_i^j	a word subsequence extracted of the sequence \mathbf{w} , formed by the words between the position i and j inclusive.
$N \times M$	number of cells in which is divided the handwritten text image. N is the number of rows and M is the number of columns or frames.
$r \times s$	number of cells on the analysis windos to obtain smoothed values of the studied features. r is the number of rows and s is the number of columns.
$n \times m$	number of pixels on the analysis windows. n is the number of rows and m is the number of columns.
e	an edge of a word graph.
$\omega(e)$	word associated with the edge e on a word graph.
q	a node on a word graph.
$t(q)$	horizontal position of the handwritten image that corresponds with the node q on a word graph.
$p(e)$	probability of the edge e on a word graph.
$\varphi(e)$	score of the edge e on a word graph.
$\phi = e_1, e_2, \dots, e_M$	a path of states on a word graph.
$S(h)$	score of the path h .
$d(\mathbf{w})$	set of paths on a word graph that produce the sequence \mathbf{w} .
$\phi_{\mathbf{w}}$	one of the paths of $d(\mathbf{w})$.
\mathbf{p}'	validated prefix of the transcription hypothesis which is error free.
v	a word of the vocabulary task.
\mathbf{p}	new validated prefix of the transcription hypothesis (\mathbf{p}' plus v).
\mathbf{s}	possible suffix that follows the validated prefix \mathbf{p} .
b	point that divide the sequence \mathbf{x} into two partes, prefix and suffix.
Q_p	set of states on a word graphs that define paths from the initial state whose associated word sequence is p .
α	grammar scale factor.
β	word insertion penalty.
(x_t, y_t)	point of the trajectory of the pen-stroke.
x'_t and y'_t	first derivatives of the point (x_t, y_t) .
x''_t and y''_t	second derivatives of the point (x_t, y_t) .
k_t	Curvature. The inverse of the local radius of the trajectory in each point.
c	class label.
M_c	a HMM for the character class c .
l_c	average length of the sequence of feature vectors used to train M_c .
f	state load factor. measures the average number of feature vectors modelled per state.
N_S	number of states on the HMMs.
N_{Sc}	number of states for the HMM character class M_c .
N_G	number of gaussians on the HMMs.

γ	weighted coefficient of the penalization due to the number of different characters between words.
e	a word transcription error.
m	a mouse action.
\mathbf{c}	a character sequence.
\mathbf{p}''	part of the validated prefix formed by completed words.
v_p	prefix of a word.
v_s	suffix of a word.
\mathbf{s}'	part of the suffix formed by completed words.
\mathbf{t}	a sequence of real value vectors representing on-line touchscreen pen strokes.
κ	a keystroke.
d	a word representing the decoding of t .

A.2 Acronyms

ASR	Automatic Speech Recognition
BIVALDI	Biblioteca Valenciana Digital
CATTI	Computer Assisted Transcription of Handwritten Text Images
CER	Character Error Rate
CS	Cristo Salvador
EER	Estimated Effort Reduction
EM	Expectation Maximisation
ER	Classification Error Rate
FKI	Research Group on computer Vision and Artificial Intelligence
GSF	Grammar Scale Factor
HMM	Hidden Markov Model
HTR	Handwritten Text Recognition
IAM	Institute of Computer Science and Applied Mathematics
IP	Interactive Predictive
IPP	Interactive Predictive Processing
IPR	Interactive Pattern Recognition
KSR	Key Stroke Ratio
LOB	Lancaster-Oslo/Bergen
MA	Mouse Action
MM-CATTI	Multimodal Computer Assisted Transcription of Handwritten Text Images
NN	Nearest Neighbour
OCR	Optical Character Recognition
OOV	Out Of Vocabulary
PKSR	Post-editing Key Stroke Ratio
PR	Pattern Recognition
WCR	Word Click Rate
WER	Word Error Rate
WG	Word Graph
WIP	Word Insertion Penalty
WSR	Word Stroke Ratio

APPENDIX *B*

Additional Experiments on Off-line Handwritten Text Recognition

Different experiments, on the three off-line corpora described in section 2, have been carried out to assess the accuracy of the off-line HTR system presented in the section 3.2. The different parameters have been optimized in order to obtain the best result for each task. On the next subsection the results obtained for the different task using n -grams at word level are detailed.

B.1 Cristo-Salvador corpora

To obtain the best result with the CS corpora we have test different values for the parameters that need to be optimized. First, we have optimized the values of the parameters N , ρ , $r \times s$ and N_S . Once this values have been fixed, the parameters N_G , α and β for the page and the book partition have been optimized independently.

The tested values for the parameters N , ρ , $r \times s$ and N_S are:

- $N = 16, 20$ and 24 .
- $\rho = 1, 2$ and 3 .
- $r \times s = 5 \times 5, 5 \times 9, 9 \times 5, 9 \times 9, 9 \times 11, 11 \times 9$ and 11×11 .
- $N_S = 6, 8, 10, 12$ and 14

As previously explained at section 3.2 automatically determination of optimal values for these parameters is not an easy task. In particular, it is difficult to determine independent, optimal values of N_S and N_G for each character HMM. For simplicity, we decided to use the same values of N_S and N_G for all HMMs.

Table B.1 shows the HTR WER(%) obtained for the corpora Cristo Salvador in its page partition for $N = 16$ and different values of the parameters ρ , $r \times s$ and N_S . Similar experiments are shown on Tables B.2 and B.3 for $N = 20$ and $N = 24$ respectively.

Table B.1: Performance of the basic off-line HTR system for $N = 16$ and different values of the parameters ρ , $r \times s$ and N_S on the CS corpora on its page partition.

N	ρ	N_S	$r \times s$						
			5×5	5×9	9×5	9×9	9×11	11×9	11×11
16	1	6	69.1	67.3	67.4	63.7	66.4	64.0	66.5
		8	83.6	79.3	80.1	78.9	78.8	78.9	79.6
		10	97.7	96.3	98.1	96.7	96.7	96.9	97.0
		12	98.6	98.1	98.6	98.1	98.1	98.6	98.2
		14	96.9	95.8	97.5	96.5	96.5	97.0	98.1
	2	6	43.5	38.5	39.2	38.5	38.2	39.9	40.9
		8	39.9	37.7	37.5	36.2	37.2	37.6	38.1
		10	39.2	36.3	35.9	36.7	37.6	38.2	38.7
		12	41.0	40.1	38.4	38.2	37.6	39.9	41.0
		14	48.0	45.5	46.1	45.1	39.7	46.1	46.0
	3	6	39.6	37.8	36.7	35.6	35.3	36.2	36.9
		8	37.2	33.6	34.3	33.1	33.4	33.8	34.6
		10	34.3	32.7	31.3	31.6	32.1	32.9	33.8
		12	31.8	31.5	30.7	30.7	30.8	31.3	32.1
		14	32.5	32.1	31.6	31.4	32.2	31.7	32.7

The best WER, 30.5%, has been obtained using $N = 20$ and $\rho = 3$. The size of the analysis windows is 9×5 and the number of states 12. After now, we are going to use this baseline to continue looking for the best value of the parameters N_G , α and β .

On Table B.4 we can see the results for the parameters values: $G_S = 32, 64, 128$; $\alpha = 60, 70, 80, 90, 100$ and $\beta = -120, -140, -160, -180$. The best result, 28.5% is obtained for $G_S = 32$, $\alpha = 90$ and $\beta = -140$.

For the CS corpora on its book partition we assume that the best values of the parameters for the feature extraction module are the same previously tuned on the CS corpora for its page partition. It is because we are using the same corpora. However, the bi-gram language model trained in this case will be different to the bi-gram trained on the page partition. So, we need to tune the parameters α and β for this partition. On Table B.5 we can see the results obtained for different values of the parameters: $N_G = 32, 64$ and 128 , $\alpha = 60, 70, 80$ and 90 and $\beta = -140, -160, -180$. The best results, 33.5% is obtained for the values $G_S = 32$, $G_{SF} = 80$ and $WIP = -160$.

The results obtained with the CS corpora in its page partition are better than the results obtained in the book partition. It is mainly due to the fact that the test text line samples of

Table B.2: Performance of the basic off-line HTR system for $N = 20$ and different values of the parameters ρ , $r \times s$ and N_S on the CS corpora on its page partition.

N	ρ	N_S	$r \times s$						
			5×5	5×9	9×5	9×9	9×11	11×9	11×11
20	1	6	50.4	48.0	44.5	44.1	45.6	44.7	46.4
		8	52.6	49.0	47.4	47.2	49.1	48.1	49.7
		10	69.2	64.6	65.9	64.0	63.6	36.9	64.8
		12	93.1	90.5	91.7	89.6	88.8	90.9	89.6
		14	98.8	97.5	98.6	97.7	97.3	97.5	96.4
	2	6	40.9	37.6	35.4	33.9	34.3	35.6	35.8
		8	35.1	35.5	32.6	37.6	32.9	33.4	33.1
		10	34.9	33.7	31.8	31.3	31.8	32.0	33.2
		12	34.3	33.3	30.9	31.7	32.2	31.7	33.1
		14	35.4	34.6	32.1	33.1	33.5	33.0	34.2
	3	6	41.4	40.7	39.3	37.7	37.8	40.3	39.9
		8	37.0	36.2	34.6	34.0	34.5	35.7	36.1
		10	35.5	34.5	32.3	32.6	32.9	33.3	34.0
		12	34.0	33.2	30.5	30.9	31.5	30.4	32.6
		14	34.9	34.5	32.0	32.1	32.0	32.8	33.4

Table B.3: Performance of the basic off-line HTR system for $N = 24$ and different values of the parameters ρ , $r \times s$ and N_S on the CS corpora on its page partition.

N	ρ	N_S	$r \times s$						
			5×5	5×9	9×5	9×9	9×11	11×9	11×11
24	1	6	44.4	39.7	36.7	35.9	37.9	36.0	38.0
		8	42.4	40.2	36.6	36.7	37.9	37.2	37.9
		10	45.8	44.3	41.1	41.5	42.8	41.4	43.5
		12	62.1	59.1	58.0	55.8	57.1	56.2	57.9
		14	85.8	82.0	84.2	80.8	79.7	80.5	79.9
	2	6	41.3	38.9	37.2	36.5	35.6	36.2	36.8
		8	38.8	36.2	34.0	34.5	33.8	32.2	33.4
		10	34.4	33.7	31.1	31.2	31.6	31.7	32.1
		12	34.6	33.9	31.2	31.7	31.6	31.5	32.0
		14	34.7	34.4	32.4	32.8	31.0	31.6	31.3
	3	6	46.6	44.5	46.6	46.3	46.0	47.1	47.3
		8	41.9	41.1	39.6	38.6	40.4	41.5	41.4
		10	38.6	37.9	36.3	36.3	37.5	37.2	38.2
		12	37.1	36.2	33.4	34.4	35.7	34.5	35.2
		14	36.2	35.0	32.8	33.2	33.2	32.9	33.1

the page partition were extracted from the same pages that the training text line samples. However, book partition better approaches a realistic transcription process as was explained

Table B.4: Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the CS corpora on its page partition.

N_G	β	α				
		60	70	80	90	100
32	-120	30.6	29.8	29.3	28.6	28.7
	-140	30.2	29.5	29.1	28.5	28.6
	-160	30.0	29.6	29.1	28.6	28.6
	-180	29.9	29.4	29.8	28.6	28.7
64	-120	31.7	31.1	30.7	30.5	30.4
	-140	31.6	31.0	30.7	30.3	30.3
	-160	31.6	30.9	30.5	30.3	30.3
	-180	31.4	30.8	30.7	30.4	30.3
128	-120	36.7	35.5	35.4	35.3	35.3
	-140	36.2	35.4	36.7	35.2	35.3
	-160	35.8	35.4	34.2	35.0	35.1
	-180	35.5	35.1	35.2	35.3	35.2

Table B.5: Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the CS corpora on its book partition.

N_G	β	α			
		60	70	80	90
32	-140	34.6	34.0	33.7	33.7
	-160	34.5	33.9	33.5	33.6
	-180	34.3	33.7	33.6	33.8
64	-140	37.3	36.9	36.0	35.9
	-160	37.2	36.5	36.1	35.9
	-180	37.0	36.6	36.1	36.0
128	-140	41.3	41.1	40.6	40.3
	-160	40.9	41.0	40.3	40.1
	-180	40.9	40.8	40.1	40.0

in Section 2.2.

B.2 ODEC corpora

Similar experiments have been carried out with the ODEC corpora. The tested values for the parameters N , ρ , $r \times s$ and N_S are:

- $N = 16, 20$ and 24 .
- $\rho = 1$ and 2 .

- $r \times s = 3 \times 3, 3 \times 5, 5 \times 3, 5 \times 5, 5 \times 9, 9 \times 5$ and 9×9 .
- $N_S = 4, 6$ and 8 .

Table B.6 show the HTR WER(%) obtained for the ODEC corpora for the different values of the parameters previously defined. The values of the parameters that obtain the best result on this table are fixed on the experiments to look for the best N_G , α and β . On Table B.7 we can see the results obtained with different values of these parameters, $N_G = 32, 64$ and 128 ; $\alpha = 10, 20, 40$ and 60 ; and $\beta = 0, -10$ and -20 . The best result (22.9%) has been obtained using $N = 16$, $\rho = 1$, $r \times s = 5 \times 3$, $N_S = 6$, $N_G = 32$, $\alpha = 20$ and $\beta = -10$.

Table B.6: Performance of the basic off-line HTR system different values of the parameters N , ρ , $r \times s$ and N_S on the ODEC corpora.

N	ρ	N_S	$r \times s$						
			3×3	3×5	5×3	5×5	5×9	9×5	9×9
16	1	4	29.6	30.7	27.5	30.8	42.9	36.0	56.2
		6	24.7	26.4	23.2	26.8	39.3	30.6	49.6
		8	41.5	41.9	41.4	42.0	48.5	46.2	57.6
	2	4	54.1	50.7	51.3	48.8	50.0	49.3	59.0
		6	32.3	32.7	35.9	34.2	43.2	36.9	42.4
		8	26.8	26.5	25.5	24.5	33.1	25.0	34.1
20	1	4	36.2	35.5	33.7	35.9	88.8	37.3	52.3
		6	27.0	27.3	24.1	26.0	45.3	27.3	41.4
		8	28.1	28.0	26.6	27.5	34.4	29.1	39.7
	2	4	75.3	73.1	68.9	43.5	62.0	70.0	63.7
		6	75.7	78.5	65.2	57.4	42.1	55.3	45.1
		8	80.1	76.3	56.7	42.3	34.0	49.7	36.3
24	1	4	48.3	45.1	45.6	42.3	48.5	50.1	54.1
		6	32.8	32.0	29.3	30.0	35.9	31.6	38.4
		8	26.6	27.2	25.3	26.0	31.6	26.8	32.9
	2	4	97.0	98.3	98.0	97.0	88.5	83.0	88.5
		6	73.4	70.2	64.8	63.0	53.5	60.3	60.0
		8	98.6	98.0	98.2	89.2	89.0	85.4	56.0

B.3 IAMDB corpora

As in the other two corpora first we have optimized the value of the parameters N , ρ , $r \times s$ and N_S , and once this values are fixed, we continue looking for the best value of the parameters N_G , α and β . Table B.8 show the HTR WER(%) obtained for the IAMDB corpora for $N = 16, 20$ and 24 ; $\rho = 1$ and 2 ; $r \times s = 3 \times 3, 3 \times 5, 5 \times 3, 5 \times 5, 5 \times 9, 9 \times 5$ and 9×9 and $N_S = 4, 6$ and 8 .

On Table B.9 we can see the results obtained for different values of the parameters N_G , α and β . The best result (25.3%) is obtained using $N = 20$, $\rho = 1$, $r \times s = 5 \times 5$, $N_S = 6$,

Table B.7: Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the ODEC corpora.

N_G	β	α			
		10	20	40	60
32	0	26.6	23.1	26.7	34.7
	-10	25.8	22.9	27.4	35.0
	-20	25.7	23.0	27.6	35.8
64	0	26.3	23.2	26.9	34.5
	-10	26.1	23.1	27.1	34.9
	-20	25.9	23.4	27.5	35.9
128	0	29.6	26.3	28.7	36.4
	-10	29.2	26.2	29.3	36.9
	-20	28.8	26.3	29.8	37.4

Table B.8: Performance of the basic off-line HTR system different values of the parameters N , ρ , $r \times s$ and N_S on the IAMDB corpora.

N	ρ	N_S	$r \times s$						
			3×3	3×5	5×3	5×5	5×9	9×5	9×9
16	1	4	36.5	36.0	36.6	35.3	40.4	40.4	49.4
		6	31.4	29.8	29.8	29.4	36.1	34.3	41.8
		8	43.0	41.8	41.3	41.0	42.8	43.2	48.6
	2	4	44.8	64.6	69.4	44.1	50.3	52.6	61.3
		6	30.9	29.8	30.0	29.8	30.8	33.4	53.8
		8	26.0	26.5	26.3	26.0	27.4	27.6	41.2
20	1	4	35.2	34.7	34.4	34.2	37.8	39.7	46.6
		6	27.5	25.8	25.5	25.3	29.0	27.1	34.9
		8	27.4	27.1	27.3	25.9	29.4	28.2	33.0
	2	4	68.2	60.5	64.9	62.3	58.8	58.6	73.2
		6	43.2	41.9	40.9	39.2	38.1	43.2	46.1
		8	37.5	37.3	34.1	33.4	34.0	35.2	38.8
24	1	4	41.1	40.0	39.8	40.6	41.3	41.9	47.8
		6	29.8	27.9	27.2	26.5	28.2	27.8	35.7
		8	29.3	27.6	26.9	25.8	27.6	26.4	34.7
	2	4	88.4	81.7	85.0	80.7	81.3	82.6	85.1
		6	73.2	65.7	65.1	63.7	65.2	64.7	68.6
		8	73.6	69.5	63.1	62.3	64.7	62.4	70.3

$N_G = 64$, $\alpha = 40$ and $\beta = 0$.

Note that the best WER obtained (25.3%) is comparable with state-of-the-art results published for this data-set [ZCB06].

Table B.9: Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the IAMDB corpora.

NG	WIP	GSF			
		20	40	60	80
32	10	32.9	28.1	34.0	39.2
	0	32.3	27.9	31.2	35.3
	-10	31.4	28.3	28.0	37.2
64	10	30.9	26.0	27.8	33.9
	0	28.8	25.3	28.3	33.6
	-10	29.1	25.6	28.1	34.2
128	10	29.0	31.1	27.9	33.1
	0	27.5	25.4	27.7	33.5
	-10	28.1	27.6	27.9	33.6

B.4 Summary

Table B.10 summarized the best results obtained with the different corpora studied in this appendix.

Table B.10: Performance of the basic off-line HTR system for different corpora.

CS-page	CS-book	ODEC	IAMDB
28.5	33.5	22.9	25.3

Bibliography

- [ZCB06] M. Zimmermann, J.-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):818–821, 2006. Member-Horst Bunke.

LIST OF FIGURES

1.1	Left: illustration of CATTI multimodal user-interaction using a touch-screen. Right: page fragment showing a line image being processed, with a partially corrected system suggestion (in grey and black roman font) and the (previous) corrections made by the user through pen strokes and handwriting input marked in red.	3
1.2	Diagram representing the different modules of an handwritten text recognition system.	5
1.3	Examples of n -grams represented using a SFSA.	16
1.4	Examples of a SFSA representing a back-off smoothed n -gram.	17
1.5	Image positions t_i are associates with the nodes as: $t(1) = t_1, t(2) = t(3) = t_2, t(4) = t_3, t(5) = t(6) = t(7) = t_4, t(8) = t(9) = t_5, t(10) = t(11) = t_6, t(12) = t_7, t(13) = t(14) = t_8, t(15) = t(16) = t(17) = t_9, t(18) = t(19) = t(20) = t_{10}$	19
2.1	Examples of the corpus “Cristo-Salvador”.	30
2.2	Detailed section of a page from the corpus “Cristo-Salvador”.	31
2.3	A sample form from the ODEC Database.	33
2.4	Examples of the answers to the suggestion point in the ODEC forms. In these examples we can see the differences on the stroke thickness, the irregular and non-consistent spacing between words and characters, the differences on the types and sizes of the words, words containing orthographic mistakes, crosses-out words, non-textual artefacts, etc.	34
2.5	Three (short) sample sentences from the ODEC Database after adequately pasting the extracted lines.	35
2.6	A sample form from the IAM Database.	36
2.7	Some sample lines from the IAM Database.	37
2.8	Some examples from the categories 1a, 1c and 1d in the Train-R01/V07 dataset	38
3.1	Overview of the HTR system.	42

3.2	Background removal, noise reduction and skew correction example: a) original image; b) skew correction, background removal, noise reductions and increase of contrast.	44
3.3	Line extraction example: Left) Image with cutting lines; Right) separated line images from the highlighted region.	45
3.4	Slant and size normalization example: Top) A separated line image; Middle) Slant correction; Bottom) Size normalization.	46
3.5	Example of the feature vectors sequence for a portion of a separated line image.	48
3.6	Example of 5-states HMM modelling (feature vectors sequences of) instances of the character “a” within the Spanish word “Castilla”. The states are shared among all instances of characters of the same class.	50
3.7	Automaton for the lexicon entry “the”.	51
3.8	A simple language model.	51
3.9	WER (%) for varying number of states per HMM and different values of the parameter N . $\rho = 2$ has been used on the left graph, and $\rho = 3$ on the right. In both cases, $r \times s = 9 \times 5$ has been used.	55
3.10	WER for different values of the parameter α and different values of the parameters β on the page partition of the CS corpora.	55
3.11	WER for different values of the parameter α and different values of the parameters β on the book partition of the CS corpora.	56
3.12	WER for different values of the parameter α and different values of the parameters β on the ODEC corpora.	57
3.13	WER for different values of the parameter α and different values of the parameters β on IAMDB corpora.	58
4.1	Overview of the CATTI system.	68
4.2	Example of CATTI interaction to transcribe an image of the Spanish sentence “antiguos ciudadanos que en Castilla se llamaban”. Initially the prefix \mathbf{p} is empty, and the system proposes a complete transcription $\hat{\mathbf{s}} \equiv \hat{\mathbf{w}}$ of the input image \mathbf{x} . In each interaction step the user reads this transcription, accepting a prefix \mathbf{p}' of it. Then, he or she types in some word, v , to correct the erroneous text that follows the validated prefix, thereby generating a new prefix \mathbf{p} (the accepted one \mathbf{p}' plus the word v added by the user). At this point, the system suggests a suitable continuation $\hat{\mathbf{s}}$ of this prefix \mathbf{p} and this process is repeated until a complete and correct transcription of the input signal is reached. In the final transcription, \mathbf{T} , the underlined italic words are the words typed by the user. In this example the estimated post-editing effort (WER) is 5/7 (71%), while the corresponding interactive estimate (WSR) is 2/7 (29%). This results in an estimated effort reduction (EER) of 59% (see Section 4.7.1 for definitions of WER, WSR and EER).	69
4.3	Example of a CATTI dynamic language model building. First, an n -gram (L) for the training set of the figure is built. Then, a linear model (L_p) which accounts for the prefix “en la” is constructed. Finally, these two models are combined into a single model ($L_p L_s$) as shown.	72

4.4	Example of edges added to a WG between the nodes i and j for probabilistic error correcting parsing. The edge labelled with the word $\omega(e)$ is the original edge and correspond to the operation of replacing the word $\omega(e)$ with itself. The group of edges labelled with $V - \{\omega(e)\}$ represent the substitution of $\omega(e)$ for another word. Here we have an edge for each word in the vocabulary except $\omega(e)$. The edge labelled with λ (empty symbol) models a deletion. Finally, the last group is for insertions, involving an edge for each word in the vocabulary from a state to itself.	75
4.5	Example of CATTI operation. Starting with an initial recognized hypothesis \hat{s} , the user validates its longest well-recognized prefix \mathbf{p}' , making a mouse-action (m), and the system proposes a new suffix, \hat{s} . As the new hypothesis does not correct the mistake the user types the corrects word v , generating a new validated prefix \mathbf{p} (v concatenated to \mathbf{p}'). Taking into account the new prefix the system suggests a new hypothesis \hat{s} starting a new cycle. Now, the user validates the longest prefix \mathbf{p}' which is error-free. The system takes into account the new prefix \mathbf{p}' to propose a new suffix \hat{s} one more time. As the new hypothesis corrects the erroneous word a new cycle start. This process is repeated until the final error-free transcription \mathbf{T} is obtained. Underlined italic word in the final transcription is the only one which was corrected by the user. Note that in the iteration 1 it is needed a mouse-click to validate the longest prefix that is error-free and then, to type the correct word. However, the iteration 2 only needs a mouse-click.	76
4.6	Example of word-graph generated after the user validates the prefix “antiguos ciudadanos que en”. The edge corresponding to the wrong-recognized word “el” was disabled.	79
4.7	Example of CATTI operation at character level. Starting with an initial recognized hypothesis \hat{s} , the user validates its longest well-recognized prefix \mathbf{p}' and corrects the following erroneous character c , generating a new validated prefix \mathbf{p} (c concatenated to \mathbf{p}'). This new prefix \mathbf{p} is submitted as additional help information to the recognition system, which based on this proposes a new suffix \hat{s} . This process goes on until the final error-free transcription \mathbf{T} is obtained. Underlined italic characters in the final transcription are those which were corrected by user.	79
4.8	Example of word-graph generated after the user validates the prefix “antiguos ci” interacting at character-level. The edge whose word does not begin with “ci” are disabled.	81
4.9	WER, WSR and EFR (all in %) for varying number of errors per sentence.	85
4.10	WER, WSR and EFR (all in %) for different values of the parameter γ	86
4.11	WER, WSR and EFR (all in %) for varying number of errors per sentence.	88
4.12	WSR, Estimated Effort-Reduction (EFR) and WCR as a function of the maximal number of MA allowed by the user before writing the correct word. The first point (0) correspond to the conventional CATTI, and the point S correspond to the single MA interaction discussed in Section 4.5.	89
5.1	Overview of the MM-CATTI system.	96

5.2 Example of multimodal CATTI interaction with a CATTI system, to transcribe an image of the Spanish sentence “antiguos ciudadanos que en Castilla se llamaban”. Each interaction step starts with a transcription prefix **p** that has been fixed in the previous step. First, the system suggests a suffix \hat{s} and the user handwrites some touchscreen text, **t**, to amend \hat{s} . This defines a correct prefix **p'**, which can be used by the on-line HTR subsystem to obtain a decoding of **t**. After observing this decoding, \hat{d} , the user may type additional keystrokes, κ , to correct possible errors in \hat{d} (and perhaps to amend other parts of \hat{s}). A new prefix, **p**, is built from the previous correct prefix **p'**, the decoded on-line handwritten text, \hat{d} , and the typed text κ . The process ends when the user enters the special character “#”. System suggestions are printed in boldface and typed text in typewriter font. User corrections are shown in red. In the final transcription, **T**, typed text is additionally underlined. Assuming all interactions as whole-word corrections, the post editing WER would be 5/7 (71%), while the MM-CATTI WSR is 3/7 (43%); i.e., 2 touch-screen + 1 keyboard word corrections. 98

5.3 Example of MM-CATTI dynamic bigram language model generation. L is the original bigram model used by off-line HTR system, whereas L_d is the bigram sub-model, derived from L , which takes as initial state that corresponding to the prefix “la”. This simplified language model is used by the on-line HTR sub-system to recognize the touchscreen handwritten word “eɔɔad”, intended to replace the wrong off-line recognized word “media”, which was disabled in L_d 100

5.4 Examples of words generated using characters from the three selected UNIPEN test writers (BH, BR, BS), along with samples of the same words written by two other writers in our labs. 101

5.5 MM-CATTI feedback decoding error rates for different writers, corpora and prefix-constrained language models. 104

6.1 System architecture. First, the web client requests a web page with an index of all available pages in the document to be transcribed. The user then navigates to a page and begins to transcribe the handwritten text images line by line. She can make corrections with pen strokes and also use the keyboard. If pen strokes are available, the MM-CATTI server uses an on-line HTR feedback subsystem to decode them. Then, taking into account the decoded word and the off-line models, the MM-CATTI server responds with a suitable continuation to the prefix validated by the user. All corrections are stored in plain text logs on the MM-CATTI server, so the user can retake them in any moment. 111

6.2 The main page of the web-based demonstrator. In addition the link to the demonstrator, a lot of information related with it is shown: videos, projects, awards, publications and teaching. 113

6.3 All the documents to transcribe. 113

6.4 By clicking on “use custom server?” link, the user can specify a custom CATTI server while her session is active. 114

6.5 A thumbnail for each page of the document chosen by the user is shown. In this case the thumbnails belong to pages of the “Cristo Salvador” book. 115

6.6 The selected page to be transcribed. 116

6.7 The HTR system proposes a full transcription of the input handwritten text line image. 116

6.8	Four interaction gestures to generate and/or validate an error-free prefix. From top to bottom: substitution, deletion, single click validation and insertion.	117
6.9	The keyboard mode.	118

LIST OF TABLES

2.1	Basic statistics of the partition <i>page</i> of the database Cristo-Salvador.	31
2.2	Basic statistics of the partition <i>book</i> of the database Cristo-Salvador.	31
2.3	Basic statistics of the database ODEC.	34
2.4	Basic statistics of the database IAM.	35
2.5	Description of the text corpora.	37
2.6	Basic statistics of the UNIPEN categories 1a, 1c and 1d in the Train-R01/V07 dataset and their corresponding partition definitions.	38
3.1	WER of the basic off-line HTR system for different values of the parameters N_S and N , fixing $\rho = 1$ and $r \times s = 5 \times 3$ on the ODEC corpora. All results are percentages.	56
3.2	WER of the basic off-line HTR system for different values of the parameters N_S and N , fixing $\rho = 1$ and $r \times s = 5 \times 5$ on the IAMDB corpora. All results are percentages.	57
3.3	WER of the basic off-line HTR system for different corpora. All results are percentages.	58
3.4	CER obtained with different n -grams at character level. All results are percentages.	59
3.5	WER obtained with different n -grams at character level. All results are percentages.	59
4.1	Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR) using the Viterbi-base search. All results are percentages.	84
4.2	Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR), excluding the sentences with zero and one post-editing errors. All reported results are percentages.	86
4.3	Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR) using the word-graph based search. All results are percentages.	87

4.4	Performance of non-interactive off-line HTR (WER) and CATTI (WSR), along with the relative difference between them (Estimated Effort-Reduction – EFR), excluding the sentences with zero and one post-editing errors using word graphs. All reported results are percentages.	87
4.5	Performance of the new single-MA interaction with CATTI system (WSR single-MA), along with Estimated Effort-Reduction for WSR single-MA with respect to WSR (EFR1) and WSR single-MA with respect to WER (EFR2). All results are percentages.	89
4.6	CER and PKSR obtained with the post-editing autocompleting approach on the HTR system. EFR for PKSR with respect to CER is also shown. All results are percentages.	90
4.7	KSR obtained with the CATTI autocompleting approach. EFR for KSR with respect to CER and KSR with respect to PKSR are also shown. All results are percentages. . .	90
5.1	Basic statistics of the UNIPEN training and test data used in the experiments.	102
5.2	For each off-line HTR task: statistics of the sets of on-line words used as feedback to correct the off-line HTR and baseline performance (classification error – ER) of the corresponding on-line HTR subsystem without using any interaction-derived contextual information (using plain 1-grams).	103
5.3	Writer average MM-CATTI feedback decoding error rates for the different corpora and three language models: plain unigram (U , <i>baseline</i>), error-conditioned unigram (U_e) and prefix-and-error conditioned bigram (B_e). The relative accuracy improvement for B_e with respect to U is shown in the last column.	103
5.4	From left-to-right: post-editing corrections (WER), interactive corrections needed (WSR), contributions of both input modalities: on-line touch-screen (TS) and keyboard (KBD), and overall estimated effort reduction (EFR) achieved by the proposed approaches. All results are percentages.	105
B.1	Performance of the basic off-line HTR system for $N = 16$ and different values of the parameters ρ , $r \times s$ and N_S on the CS corpora on its page partition.	138
B.2	Performance of the basic off-line HTR system for $N = 20$ and different values of the parameters ρ , $r \times s$ and N_S on the CS corpora on its page partition.	139
B.3	Performance of the basic off-line HTR system for $N = 24$ and different values of the parameters ρ , $r \times s$ and N_S on the CS corpora on its page partition.	139
B.4	Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the CS corpora on its page partition.	140
B.5	Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the CS corpora on its book partition.	140
B.6	Performance of the basic off-line HTR system different values of the parameters N , ρ , $r \times s$ and N_S on the ODEC corpora.	141
B.7	Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the ODEC corpora.	142

B.8 Performance of the basic off-line HTR system different values of the parameters N , ρ , $r \times s$ and N_S on the IAMDB corpora. 142

B.9 Performance of the basic off-line HTR system for different values of the parameters α , N_G and β on the IAMDB corpora. 143

B.10 Performance of the basic off-line HTR system for different corpora. 143

