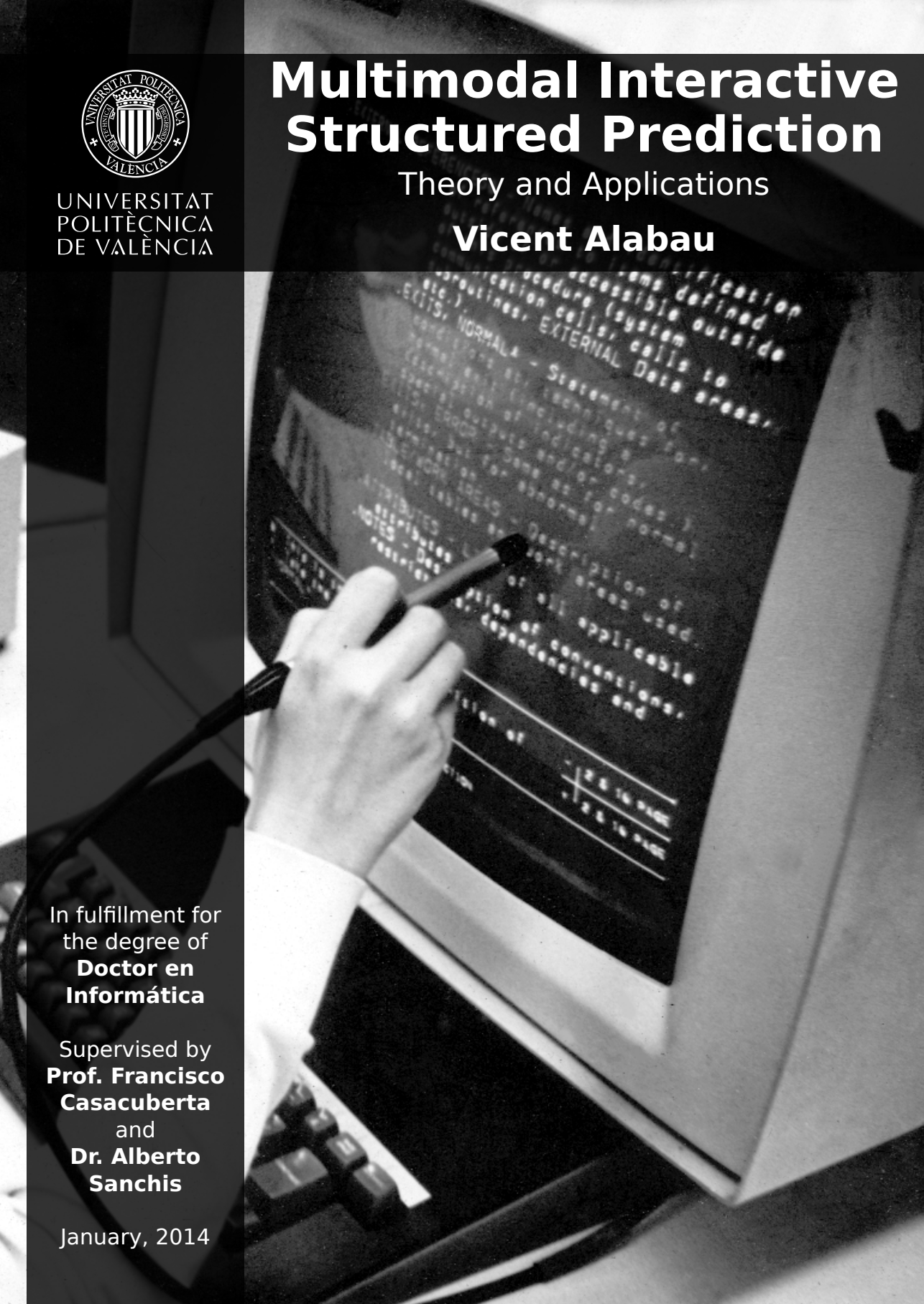


Multimodal Interactive Structured Prediction

Theory and Applications

Vicent Alabau

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

In fulfillment for the degree of **Doctor en Informática**

Supervised by **Prof. Francisco Casacuberta** and **Dr. Alberto Sanchis**

January, 2014

Universitat Politècnica de València
Departament de Sistemes Informàtics i Computació

Multimodal Interactive Structured Prediction

Theory and Applications

by Vicent Alabau

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor en Informática

supervised by
Prof. Francisco Casacuberta and Dr. Alberto Sanchis

January, 2014



The research leading to this thesis has been partially funded by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287576 (CasMaCat) and by the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018). Also, it has been supported by the "Vicerrectorado de Investigación de la UPV" under the "Programa d'Incentiu a la Investigació 2004 UPV".

Document prepared and typeset in LaTeX.
Cover design by Vicent Alabau. Original photo of the Hypertext Editing System console in use at Brown University by Greg Lloyd. Retrieved from Wikimedia Commons with license CC-BY-2.0.

Board Committee

President .....

Prof. Enrique Vidal
Universitat Politècnica de València

Vocal .....

Prof. Marcello Federico
Fondazione Bruno Kessler

Secretary .....

Prof. Philipp Koehn
University of Edinburgh

Valencia, January, 2014

Acknowledgements

Volia aprofitar estes línies per a agrair a totes les persones que, d'una o altra forma, m'han oferit el seu suport i confiança per a que esta tesi s'haja pogut acabar.

En primer lloc, voldria agrair sincerament a tots aquells que han sigut una referència a l'hora de formar-me com a investigador. En especial als meus directors de tesi, Paco i Alberto, i al que ho va ser en una etapa inicial, Carlos, pel seu suport al llarg dels anys. Ells varen confiar en mi i em donar l'oportunitat de entrar en este món tan apassionant. També voldria agrair a Alfons i a Enrique, amb els quals vaig fer el projecte fi de carrera, per introduir-me en l'àrea de reconeixement de formes i reclutar-me al grup PRHLT. Tots ells, junt als altres professors del PRHLT, m'han ensenyat la importància de la rigorositat científica, el desenvolupament teòric, i en resum, tot allò qué suposa la professionalitat al món científic. I also would like to thank Prof. Hermann Ney for welcoming me in his research group for three months. During that period I had the chance to learn how is organized one of the top research groups in pattern recognition. Thanks to David Vilar and Daniel Stein for making my stay in Aachen much more enjoyable. Of course, I also would like to thank Prof. José Oncina, Prof. Philippe Langlais and Prof. Philipp Koehn for their kind reviews and valuable comments regarding the draft of this thesis, and Prof. Enrique Vidal, Prof. Marcello Federico and Prof. Philipp Koehn (again) for being members of the board committee. Special thanks to José Oncina, whose work has been a great inspiration for many of the work carried out in this thesis.

Al llarg d'estos anys com a informàtic i com investigador he anat trobant companys d'estudis i de feina que han acabat esdevenint amics. Tot va començar amb els Jailers, amb qui vaig descobrir que la informàtica podia ser una cosa molt divertida. Anys després, el laboratori L105 va resultar ser no sols un lloc on creixer professionalment sinó també un lloc on creixer personalment. Gràcies a Neus, Stella, Maite, Vero i Miriam per escoltar les meues penes, i per l'afecte que m'han donat. A German, Jorge, Guillem i la resta del laboratori per compartir les rises i els bugs del compilador. Més endavant, vaig acabar a l'ITI junt a alguns companys com Vero i German. Allí vaig obrir una nova etapa

professional amb el projecte MIPRCV i ha continuat amb CasMaCat. Este periode està siguent especialment productiu a l'hora de nutrir-me de l'experiència d'altres investigadors. Gràcies a Alejandro, Jose Ramón, Antonio, Dani, entre altres, per compartir les seues experiències. Especialment he d'agraïr a Luis, qui té una capacitat de treball extraordinaria, i que m'ha aportat una forma distinta de vore la investigació gracies a l'atenció als detalls, el gust pel coneixement transversal, i per la seua professionalitat. Voldria tornar a agrair a Vero pel seu continuat suport i per compartir el humor i el cant a dos veus.

Per últim i no per això menys important, he d'agrair a la meua família: la de sempre, l'elegida, l'adoptiva i la futura. Ma mare, que en pau descanse, no va poder vore començar la meua etapa investigadora, com tampoc ho va fer d'altres esdeveniments importants que han passat després de la seua mort. No obstant això, estic convençut de que estaria molt orgullosa de l'evolució dels quatre membres de la família. A mon pare, a Javier i a Eva, que perdonen la falta de dedicació que els he pogut tindre. No obstant això, han estat sempre quan ha fet falta. Als Alabaus en general pels jocs i les discussions intel·lectuals, i als Gonzalvos pel calor humà i el 'bon menjar'. Gràcies a Elsa i Miquel, els quals sent com a part de la meua família. Malgrat alguns mals moments que hem passat, sempre hem acabat tornant a una amistat reforçada. Ambdós són les persones que més han influït en la definició de la meua personalitat i els tinc un deute etern. Gracies també als amics d'Alzira i Mislata. En els darrers anys, la familia Martínez Zapata ha sigut la meua familia d'acollida. Gracies a Mari Paz per cuidar-me com a un fill, a Antonio per 'chincharme' i a Hugo i Iker per eixos passejos matutins amb llançament de pedres inclòs que em permeten possar les idees en ordre i desestresar-me. I per damunt de tot a Maria, qui comença a entrendre'm com ningú. Amb amor, em fa vore el que jo no sóc capaç de vore, i m'orienta i m'ajuda a disciplinar-me quan em desvie del camí correcte. Valors necessaris per a establir la futura família.

Vicent Alabau
January, 2014

Abstract

This thesis presents scientific contributions to the field of multimodal interactive structured prediction (MISP). The aim of MISP is to reduce the human effort required to supervise an automatic output, in an efficient and ergonomic way. Hence, this thesis focuses on the two aspects of MISP systems. The first aspect, which refers to the interactive part of MISP, is the study of strategies for efficient human–computer collaboration to produce error-free outputs. Multimodality, the second aspect, deals with other more ergonomic modalities of communication with the computer rather than keyboard and mouse.

To begin with, in sequential interaction the user is assumed to supervise the output from left-to-right so that errors are corrected in sequential order. We study the problem under the decision theory framework and define an optimum decoding algorithm. The optimum algorithm is compared to the usually applied, standard approach. Experimental results on several tasks suggests that the optimum algorithm is slightly better than the standard algorithm.

In contrast to sequential interaction, in active interaction it is the system that decides what should be given to the user for supervision. On the one hand, user supervision can be reduced if the user is required to supervise only the outputs that the system expects to be erroneous. In this respect, we define a strategy that retrieves first the outputs with highest expected error first. Moreover, we prove that this strategy is optimum under certain conditions, which is validated by experimental results. On the other hand, if the goal is to reduce the number of corrections, active interaction works by selecting elements, one by one, e.g., words of a given output to be supervised by the user. For this case, several strategies are compared. Unlike the previous case, the strategy that performs better is to choose the element with highest confidence, which coincides with the findings of the optimum algorithm for sequential interaction. However, this also suggests that minimizing effort and supervision are contradictory goals.

With respect to the multimodality aspect, this thesis delves into techniques to make multimodal systems more robust. To achieve that, multimodal systems are improved by providing contextual information of the application at hand. First, we study how to integrate e-pen interaction in a machine translation task. We contribute to the state-of-the-art by leveraging the information from

the source sentence. Several strategies are compared basically grouped into two approaches: inspired by word-based translation models and n-grams generated from a phrase-based system. The experiments show that the former outperforms the latter for this task. Furthermore, the results present remarkable improvements against not using contextual information. Second, similar experiments are conducted on a speech-enabled interface for interactive machine translation. The improvements over the baseline are also noticeable. However, in this case, phrase-based models perform much better than word-based models. We attribute that to the fact that acoustic models are poorer estimations than handwritten morphologic models and, thus, they benefit more from the language model. Finally, similar techniques are proposed for dictation of handwritten documents. The results show that speech and handwritten recognition can be combined in an effective way.

Finally, an evaluation with real users is carried out to compare an interactive machine translation prototype with a post-editing prototype. The results of the study reveal that users are very sensitive to the usability aspects of the user interface. Therefore, usability is a crucial aspect to consider in an human evaluation that can hinder the real benefits of the technology being evaluated. Hopefully, once usability problems are fixed, the evaluation indicates that users are more favorable to work with the interactive machine translation system than to the post-editing system.

Resumen

Esta tesis presenta una serie de contribuciones científicas en el campo del reconocimiento de formas interactivo y multimodal (MISP, del inglés "multimodal interactive structured prediction"). El objetivo de MISP es reducir el esfuerzo humano necesario para corregir la salida de un sistema automático, de forma eficaz y ergonómica. La tesis se centra en los dos aspectos principales de MISP: interacción y multimodalidad. El objetivo de la interacción es el estudio de estrategias de colaboración hombre-máquina para producir resultados sin errores. La multimodalidad trata de cómo utilizar modalidades de comunicación con la máquina más ergonómicas que las tradicionales (teclado y ratón).

En la interacción secuencial (de izquierda a derecha), el usuario supervisa la salida automática secuencialmente, de forma que los errores son corregidos en orden de aparición. En esta tesis, el problema de la interacción secuencial se estudia bajo el marco de la Teoría de la Decisión, bajo la cual se define un algoritmo óptimo. Los resultados experimentales sobre varias tareas sugieren que el algoritmo óptimo consigue mejores resultados que el algoritmo estándar.

En comparación con la interacción secuencial, donde el usuario toma la iniciativa, en la interacción activa es el sistema quien decide lo que el usuario tiene que supervisar. Por un lado, se puede reducir el esfuerzo pidiendo al usuario que revise sólo las salidas automáticas susceptibles de tener errores. Para ello, se define una estrategia que pide las salidas con un mayor error esperado en primer lugar. A continuación, se demuestra y valida experimentalmente que esta estrategia es óptima bajo ciertas condiciones. Por otro lado, si el objetivo es reducir el número de correcciones, la interacción activa funciona seleccionando, elemento a elemento, qué elemento de la salida tendría que ser supervisada por el usuario, por ejemplo, una palabra. En este caso, se comparan diversas estrategias. Al contrario que en el caso anterior, la estrategia que funciona mejor aquí es la de escoger el elemento en el que tenemos una confianza mayor, lo que coincide con los resultados del algoritmo óptimo para la interacción secuencial. Esto sugiere que minimizar el esfuerzo y la supervisión son objetivos que van en dirección opuesta.

Con respecto a la multimodalidad, esta tesis profundiza en técnicas para conseguir sistemas multimodales más robustos. Para ello, los sistemas multimodales

se mejoran de forma que puedan aceptar información contextual de la aplicación que se está utilizando. En primer lugar, se estudia cómo integrar el lápiz electrónico en una tarea de traducción automática. Los modelos de lenguaje se mejoran aprovechando la información de la frase origen. Se comparan varias estrategias que se pueden agrupar en las siguientes aproximaciones: una inspirada en los modelos de traducción basados en palabras y otra basada en sistemas de traducción basados en segmentos. Los resultados demuestran una mejoría notable respecto a las estrategias habituales que no utilizan información contextual. La voz también puede ser una modalidad de entrada interesante, dado que deja las manos libres para otras tareas. En este sentido, se prueban experimentos similares a los del lápiz electrónico pero para una interfaz que permite interaccionar con la voz. Las mejoras sobre la aproximación estándar son importantes. Cuando se compara con las estrategias del lápiz electrónico, los modelos basados en frases consiguen mejor rendimiento que modelos basados en palabras. Este hecho es atribuible a que los modelos acústicos están probablemente peor estimados que los modelos morfológicos de escritura y, por eso, se benefician más del modelo de lenguaje. Finalmente, se proponen técnicas similares para el dictado de documentos manuscritos. Los resultados demuestran que el reconocimiento de voz y el reconocimiento de textos manuscritos se pueden combinar de una forma eficaz.

Por último, la tesis presenta una evaluación con usuarios reales de un prototipo de traducción interactiva que se compara con un sistema de post-edición de la traducción. Los resultados del estudio revelan que los usuarios son muy sensibles a aspectos de usabilidad de la interfaz de usuario. Por tanto, la usabilidad es un aspecto crucial a considerar en una evaluación humana ya que puede impedir que los usuarios aprecien los beneficios reales de la tecnología. Por suerte, si se corrigen los problemas de usabilidad, el estudio indica que los usuarios prefieren trabajar con el sistema de traducción interactivo a usar el sistema de post-edición.

Resum

Esta tesi presenta una sèrie de contribucions científiques al camp del reconeixement de formes interactiu i multimodal (MISP, de l'anglès "multimodal interactive structured prediction"). L'objectiu de MISP és reduir l'esforç humà necessari per corregir l'eixida d'un sistema automàtic, de manera eficaç i ergonòmica. La tesi es centra en els dos aspectes principals de MISP: interacció i multimodalitat. L'objectiu de la interacció és l'estudi d'estratègies de col·laboració home-màquina per tal de produir resultats sense errors. La multimodalitat tracta de com utilitzar modalitats de comunicació amb la màquina més ergonòmiques que les tradicionals (teclat i ratolí).

En la interacció seqüencial (d'esquerra a dreta), l'usuari supervisa l'eixida automàtica seqüencialment, de manera que els errors són corregits en l'ordre d'aparició. En esta tesi, el problema de la interacció seqüencial s'estudia baix el marc de la Teoria de la Decisió, baix la qual es defineix un algoritme òptim. Els resultats experimentals sobre diverses tasques suggereixen que l'algoritme òptim aconsegueix millors resultats que l'algoritme estàndard.

En comparació amb la interacció seqüencial, on l'usuari pren la iniciativa, en la interacció activa és el sistema qui decideix el que l'usuari ha de supervisar. D'una banda, es pot reduir l'esforç demanant a l'usuari que revise només les eixides automàtiques susceptibles de tindre errors. Per a això, es defineix una estratègia que demana, en primer lloc, les eixides amb un major error esperat. A continuació, es demostra i valida experimentalment que esta estratègia és òptima baix certes condicions. D'altra banda, si l'objectiu és reduir el nombre de correccions, la interacció activa funciona seleccionant, element a element, quin element de la eixida hauria de ser supervisada per l'usuari, per exemple, una paraula. En este cas, es comparen diverses estratègies. Al contrari que en el cas anterior, l'estratègia que funciona millor ací és la d'escollir l'element amb el qual tenim una major confiança, cosa que coincideix amb els resultats de l'algoritme òptim per a la interacció seqüencial. Això suggereix que minimitzar l'esforç i la supervisió són objectius que van en direcció oposada.

Respecte a la multimodalitat, esta tesi aprofundeix en tècniques per aconseguir sistemes multimodals més robustos. Per això, els sistemes multimodals es milloren de manera que puguen acceptar informació contextual de l'aplicació

que s'està utilitzant. En primer lloc, s'estudia com integrar el llapis electrònic en una tasca de traducció automàtica. Els models de llenguatge es milloren aprofitant la informació de la frase origen. Es comparen diverses estratègies que es poden agrupar en les següents aproximacions: una inspirada en els models de traducció basats en paraules i una altra basada en sistemes de traducció basats en segments. Els resultats demostren una millora notable respecte a les estratègies habituals que no utilitzen informació contextual. La veu també pot ser una modalitat d'entrada interessant, atès que deixa les mans lliures per a altres tasques . En este sentit, es proven experiments similars als del llapis electrònic però per a una interfície que permet interaccionar amb la veu. Les millores sobre l'aproximació estàndard són importants. Quan es compara amb les estratègies del llapis electrònic, els models basats en frases aconsegueixen millor rendiment que models basats en paraules. Este fet és atribuïble a que els models acústics estan probablement pitjor estimats que els models morfològics d'escriptura i, per això, es beneficien més del model de llenguatge. Finalment, es proposen tècniques similars per al dictat de documents manuscrits. Els resultats demostren que el reconeixement de veu i el reconeixement de texts manuscrits es poden combinar d'una manera eficaç.

Finalment, la tesi presenta una avaluació amb usuaris reals d'un prototip de traducció interactiva que es compara amb un sistema de post-edició de la traducció. Els resultats de l'estudi revelen que els usuaris són molt sensibles als aspectes d'usabilitat de la interfície d'usuari. Per tant, la usabilitat és un aspecte crucial a considerar en una avaluació humana ja que pot impedir que els usuaris aprecien els beneficis reals de la tecnologia. Per sort, si es corregeixen els problemes d'usabilitat, l'estudi indica que els usuaris prefereixen treballar amb el sistema de traducció interactiu a utilitzar el sistema de post-edició.

Preface

Structured prediction (SP) is a classification problem in which the output consists of structured labels (as opposed to independent labels), i.e. the labels in the output have dependencies among them. Examples of structured outputs are natural language, DNA sequences or an XML describing the layout analysis of a web page. Traditionally, SP has been approached as a fully automated procedure. The automatic SP scenario can be described as follows: an input is presented to a SP system. Then the SP system produces an output, which will typically contains some errors. Finally, if a "perfect" or a high-quality result is desired, it is convenient for a human expert to revise the system output. The purpose of the expert is to amend the errors to produce the final output (or reference). This process is known as *post-editing* (PE).

The previous scheme has been adopted for many SP problems, to the point that SP algorithms have specialized in minimizing the number of errors produced. Nevertheless, in PE all the effort in correcting the output is delegated to the user, which is not desirable as human labor is expensive. Then, how can we increase the productivity of the human responsible of ensuring a high quality output? As in the manufacturing industry, where machines perform heavy duty tasks and humans fill the gap of what machines cannot yet do, in *interactive structured prediction* (ISP), systems are responsible for efficiency and productivity, whereas the human user must deal with correctness and quality control.

In this thesis we tackle this problem from two perspectives: how the interaction can be performed efficiently and the way interaction can be approached. For the former, we can consider to cases: passive interaction, where the user takes the initiative to supervise and correct the system output; and active interaction, where it is the system that decides what output should be supervised and how. For the latter, the interaction can be achieved by means of the keyboard and mouse, in a classical set-up, or by means of other arguably more efficient or ergonomic modalities such as speech or hand-writing.

Typically, search algorithms for passive ISP problems have been based on the algorithms for fully-automatic SP systems. However, the decision rule applied should not be considered as optimal since the goal in ISP is to reduce human

effort instead of output errors. To this respect, this work aims to give insight into the optimal decision rule for ISP problems, find efficient algorithms and asses the proposed methods with real world natural language processing tasks.

In the previous interaction scheme, the user must revise the whole output to achieve the desired level of quality. Nonetheless, in the case that the amount of data is so vast that a thorough supervision becomes prohibitive, the user can rely on the system to retrieve the outputs that are likely to have more errors. Then, the effort can be directed to supervise only a subset of the data worthwhile supervising, with the goal to minimize the residual error. We call this active interaction.

Additionally, since the breakout of tactile smartphones, the number of devices featuring a touch-screen has been increasing at a fast pace. The success of tactile smartphones has fostered a new kind of keyboardless technology which was latent until then: the tablet computers. They have been presented as a substitute of paper notebooks, as they have a similar size. Nevertheless, the possibilities this new technology may provide are still to be unveiled. In that context, on-line handwritten text recognition (HTR) plays a crucial role. First, because to input text in such devices using a virtual keyboard is far from the efficiency of regular keyboards. Secondly, handwriting is a natural way to communicate. Withal, a HTR interface can commit recognition errors. Thus, if the HTR system is not robust enough, user experience could be negatively affected hindering its use.

The problem of integrating speech in interactive systems is similar to that of integrating e-pen. Interacting with speech may seem not as useful as with e-pen, since the speech error rates are a bit higher and it is difficult to picture, for instance, a professional translator/transcriptor speaking the whole corrections aloud. However, speech interaction is still an interest approach. That is the case of persons with disabilities, that cannot perform their work in the usual way, as they have trouble using their hands. Instead, they can still perform their duties using speech. Another interesting case is when speech can be used along with the keyboard or the e-pen. Here, speech allows the user not to lose the focus on her work and allows her to keep the hands in the keyboard. Finally, it is not difficult to imagine, for instance, a paleographer reading aloud a handwritten ancient book just for fun or to illustrate others. This reading could be leveraged to perform or improve the transcription of such texts.

Thus, the scientific goals of this thesis can be summarized in the following contributions:

1. **Algorithms for passive interaction (Chapter 3).** First, it is described as a *decision theory* problem from which a general analytical formulation of the optimum algorithm is derived. Then, it is compared with the standard formulation to establish under which conditions the standard algorithm should perform similarly to the optimal algorithm.

2. **Algorithms for active interaction (Chapter 4).** In addition, an active interactive scenario is studied, in which it is the machine that proposes a structure or label to correct. The first attempts to reduce the number of supervisions whereas the second aims at reducing the number of correction the user needs to make. Based on active learning techniques, several search algorithms are explored and compared.

3. **E-pen interaction (Chapter 5).** In the second part of this work we research on how to integrate alternative modalities for interacting with structured prediction systems. On the one hand, current technology for on-line handwriting text recognition (HTR) is far from developing error-free systems. Consequently, its use in many applications is limited. To this respect, we have developed an on-line HTR system that leverages the information in interactive machine translation (IMT). Empirical experimentation suggests that this information can be used efficiently to improve the robustness of the on-line HTR system, achieving remarkable results.

4. **Speech interaction (Chapter 6).** On the other hand, we present a new approach to perform speech interaction in a way that translation and speech inputs are tightly fused. This integration is performed early in the speech recognition step. Thus, the information from the translation models allows the speech recognition system to recover from errors that otherwise would be impossible to amend. In addition, this technique allows to use currently available speech recognition technology. The proposed system achieves an important boost in performance with respect to previous approaches.

5. **Design and evaluation of a web-based prototype (Appendix A).** Finally, we present a multi-user web-based prototype for IMT which has been assessed by human evaluators. We report the lessons learned from two user evaluations. Our results can provide researchers and practitioners with several guidelines towards the design of on-line IMT tools. In addition, the results suggest in which direction future research efforts should be driven.

Contents

Acronyms

AISP active interaction for structured prediction

API application programming interface

ASR automatic speech recognition

AUC area under the curve

CER classification error rate

FSM finite-state machines

HCI human–computer interaction

HMM hidden Markov model

HTR handwritten text recognition

KSR key stroke ratio

MAP maximum-a-posteriori

MBR minimum Bayes risk

MCE minimum classification error

MIPRCV Multimodal Interaction in Pattern Recognition and Computer Vision

MT machine translation

OCR optical character recognition

PE post-editing

PR pattern recognition

QE quality estimation

SISP sequential interactive structured prediction

SP structured prediction

WER word error rate

WG word graph

WSR word stroke ratio

“A year passed: winter changed into spring, spring changed into summer, summer changed back into winter, and winter gave spring and summer a miss and went straight on into autumn... until one day...”

Monty Python and the Holy Grail

Chapter 1

Preliminaries

Chapter Outline

1.1 Preamble

Today, in the Information Age, a vast amount of data is generated on daily basis. According to Google's CEO Eric Schmidt "every two days now (2010) we create as much information as we did from the dawn of civilization up until 2003"[1]. The amount of data generated is so vast that traditional manual methods cannot be used to analyze the data efficiently and make sense of it. First, because processing the data with human labor would be very expensive. Second, because some tasks can be performed more efficient when machines come into play, as it was observed during the Industrial Revolution [Wells, 1899]. Thus, computers can be a huge asset to this respect by converting data into usable information. However, these data often come in a form that cannot be directly addressed by the computer. This is the case when the data is extracted from sensory signals, as the result of a digitalization of a physical process, e.g., images, audio or video [Gorman, 2003]. In other occasions, the data is already in a more computer-friendly representation, such as blog entries, short messages. Even so, these data are the product of a natural human process and it is not the kind of the structured unambiguous data that computers are able to manage efficiently, i.e., natural human processes are riddled with ambiguities.

1.1.1 Statistical pattern recognition

As it was defined in Liu et al. [2006], the fundamental role of *pattern recognition* (PR) is "executing the tasks like human being on computers". Hence, PR has become very valuable since it allows automatic interpretation of the input data. This is achieved by assigning a label or set of labels to the input data. In supervised PR, the labels are predefined, and thus, the meaning of the labels is known. Hence, the interpretation of the input is achieved by the interpretation of the labels. In supervised statistical PR [Jain et al., 2000], which is the approach that we will use in this thesis, the decoding of ambiguous data into labels is treated as a stochastic problem. We can identify three principal stages:

Data acquisition and preprocessing. The data is converted into a format that the computer can understand while trying to preserve the critical information necessary to interpret the input. Typically, a set of features that are considered to be relevant to the problem at hand are extracted from the data.

Data representation and learning. In order to represent the data, an appropriate statistical model must be chosen. The models are defined by a set of statistical parameters that are estimated based on a set of training (labeled) data.

[1] http://techcrunch.com/2010/08/04/schmidt-data/

Decision making. When a new input is observed, a classification algorithm is used to decide which output labels are to be assigned, on the basis of the statistical parameters obtained in training.

Typically, researchers have focused on improving any of these stages, since they all are equally important to design a robust and accurate PR system. Unfortunately, although PR has shown to achieve good results on many tasks, the accuracy of such systems is usually not good enough to mimic human performance. Thus, full automation is still not possible when human-like accuracy is desired. In that case, human operators are needed to supervise the system's output and to amend possible errors. This process is typically carried out separately from the PR process, just as a manual work, and as such, it can be deemed as inefficient, expensive, and tedious.

Multimodal interactive PR [Toselli et al., 2011] is a paradigm for PR where a human operator takes part in the **decision making** stage by giving some feedback to guide the system towards a high quality solution. This process is performed iteratively until the human operator agrees with the final quality of the output. Three aspects are crucial for this paradigm to succeed:

Feedback is related to how the human–computer *interaction* is performed effectively and efficiently to increase the productivity in the generation of correct solutions.

Multimodality deals with how computer and humans communicate, that is, which *modalities* can be used to perform the interaction more ergonomically and how they are used.

Adaptation relates to how systems can *learn* from data derived from user interactions to improve the system response in the future, and tune it to the user behavior and the specific task considered.

This thesis is devoted to the study of the **feedback** and **multimodality** aspects of multimodal interaction. On the one hand, we aim at finding algorithms that can perform such interaction optimally. On the other hand, we will explore different alternatives to integrate ambiguous modalities, such as handwriting and speech, into the interactive system. Finally, we will validate our assumptions regarding the interaction protocols by testing an interactive system with real users.

Nevertheless, it is out of the scope of this thesis to cover the **adaptation** aspect of multimodal interaction. The interested reader is referred to [Nepveu et al., 2004; Rodríguez-Ruiz, 2010] for short term adaptation and to [Martínez-Gómez et al., 2011; Ortiz-Martínez, 2011] for long term adaptation for interactive systems. For a general overview of the multimodal interaction paradigm see [Toselli et al., 2011].

1.1.2 Human–computer interaction in PR

Coined by Carlisle [1976] and popularized by Card et al. [1980] in the eighties, *human–computer interaction* (HCI) [Dix et al., 2004; Jacko and Sears, 2007] is a term used to describe the study, planning, and design of the interaction between users and computers. The goal of HCI is to improve the interaction between humans and computers by designing interfaces, developing new interaction techniques and evaluating them to prove their effectiveness. This broad definition of HCI is the umbrella for many research communities. Particularly, the interests of this thesis regarding HCI are focused on how to improve user productivity and output quality of a PR system by leveraging human–computer interaction.

HCI and PR have a long history in common. Unsurprisingly, *machine translation* (MT) was one of the first problems to adopt interactive technologies, as it is a very difficult problem that can benefit much from user's feedback. The first studies date back to 1970, where Kay and Martins [1970] proposed a question/answer interface in which the user's role was to determine word sense, phrasal attachments, etc. in the source text. The information gathered was used to improve the translation quality. That work was followed by others [Brown and Nirenburg, 1990; Whitelock et al., 1986], but the method proved to require well trained experts, and therefore it was not very practical. In addition, the user was not in control of the translation process, but she was only queried regarding some aspects of the source text. It was not until the TransType project that Foster et al. [1996] proposed that the interaction should shift from focusing on the source text meaning to address directly the final target translation. The goal was to put the user in control of the production of the translation, instead of a manual review and correction of a draft translation. The system helped the user by providing useful word auto-completion that was sensitive to the source text. Later on, that work was extended to provide auto-completion of several words. Furthermore, a follow-up project TransType2 [Esteban et al., 2004] came with several technological enhancements that improved the prediction capabilities and usability of the interfaces [Barrachina et al., 2009]. A representation of a TransType2 kind of interaction is shown in Figure 1.1. In this illustration, the user translates the sentence "Para imprimir una lista de fuentes postscript:" in Spanish into the sentence "To print a list of postscript fonts:" in English. Each time the user fixes the translation by changing a word, the system responds with a suffix sentence as in auto-completion. This is one of the interactive scenarios that we will study throughout this thesis.

The work carried out in TransType2 fostered a new kind of predictive technologies (in the sense of auto-completion) for other PR problems, namely, transcription of text images [Toselli et al., 2010], predictive parsing [Sánchez-Sáez et al., 2009], speech transcription [Rodríguez et al., 2007] and text generation [Rodríguez et al., 2010]. A slightly different approach for interaction

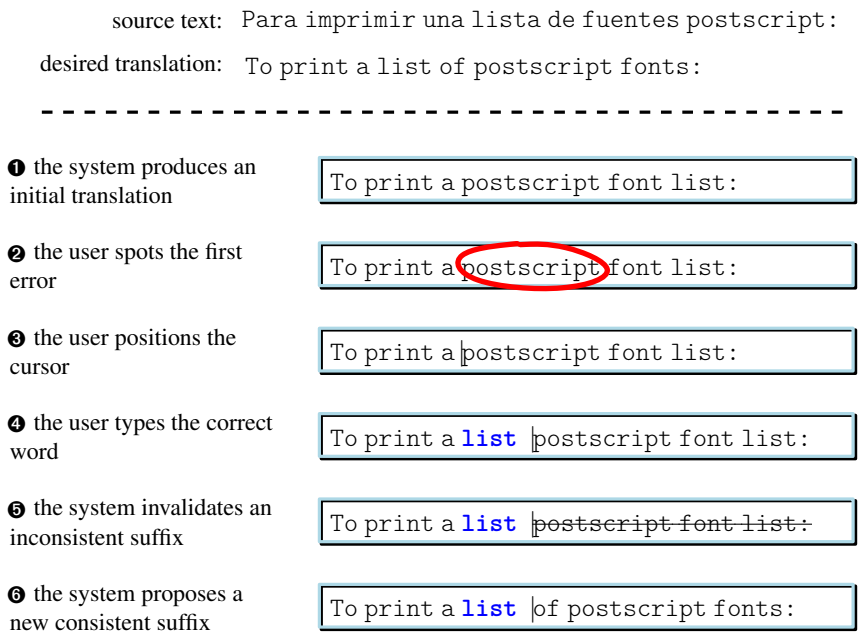


Figure 1.1: Example of a TransType2 kind of interaction for translating "Para imprimir una lista de fuentes postscript:" in Spanish into "To print a list of postscript fonts:" in English. As the user does not like the first system translation, she positions the cursor to introduce the changes. Then, the system takes into account the user's suggestion to produce a suffix that is more consistent with the new information provided. The interaction is conceived as an auto-complete feature.

with MT systems was studied in [Koehn, 2009], where the translation interface supported not only translation auto-completions but also presented selectable phrase translation options. Alternatively, Culotta et al. [2006] studied how user feedback could be used to correct the prediction of *conditional random fields* to solve problems like named-entity recognition. While they named their technique *corrective feedback*, the concepts were quite similar to what was proposed in TransType2.

Another related, but less PR centric, approach is what Horvitz [1999] called *mixed-initiative user interfaces*. In that work, a set of principles are described on how intelligent agents should interact with users. In their opinion, user interfaces should enhance users' abilities to directly manipulate objects. Instead of providing full automation, the interfaces should provide mechanisms for user-computer collaboration to refine the results. The automation provided by the computer should also add significant value. As it can be observed, the shift in the paradigm proposed by TransType fits in the definition of Horvitz *mixed-initiative user interfaces*: the user interacts with the system by manipulating directly the output whereas the system enhances user productivity with

smart auto-completion. Afterwards, Shilman et al. [2006] connected Horvitz [1999] ideas with the work performed in [Culotta et al., 2006]. They designed an interface that allowed users to correct on-line handwritten text by directly transforming the recognized text. The interaction technique was very similar to that derived from TransType2.

In all those techniques, it is the user who takes the initiative in proposing a correction. However, the system can also take the initiative and ask the user for a specific feedback. This kind of scenario is especially valuable since it allows the user to avoid supervising the whole output. Instead, the user is given hints of what may need supervision. Therefore, if the system is precise enough the user saves effort and, at the same time, the quality of the output improves. This type of *active* interaction between humans and PR systems is related to *active learning* (AL)[2] [Settles, 2010], where a great deal of research effort has been addressed. The goal of AL systems is to ask the user to label data so as to generate training samples. By doing this, AL aims to build better models with less data, i.e., to achieve more accurate models requiring less user effort in labeling the data. Note that, although related, AL and *active* interaction aim at different purposes. In *active* interaction, our interests focus on how to produce high-quality output from state-of-the-art PR systems. Thus, we will assume that we already have state-of-the-art feature extraction, and that we possess enough training data to appropriately train our models. Hence, we expect that the accuracy of the PR system has little to improve from more training data, except for out-of-domain data. Fortunately, interaction techniques similar to what AL proposes can be adapted to our particular problem. For instance, Oncina and Vidal [2011] used an *active interaction* technique to improve the output of a chromosome classification problem. In the active interaction scenario, the system proposed what labels to correct, achieving less corrections than in a classical interaction scheme. In [Culotta et al., 2006], active interaction was also evaluated, where the user was asked to fix the *lest confident* label instead of letting the user select a random label. However, in their experiments the active version did not obtain significant improvements. Moreover, another kind of active interaction is Serrano et al. [2010] and González-Rubio et al. [2010] approach for balancing error and supervision effort. In those works, the word confidences were used to direct user's attention towards the parts of the sentence that needed corrections. The results were quite encouraging since user effort could be reduced significantly while the transcription or translation error was kept in a reasonable level.

Many other works deal with some sort of interaction with PR systems. For instance, spam classification is a popular feature in e-mail applications. They usually implement uncomplicated interaction techniques to correct the system prediction: a button to tell if an e-mail was actually spam or not [Ramachandran et al., 2011]. As spam detection is a binary classification problem, correcting the output consists simply in choosing the right label. The same can

[2] Also known as *query-based learning*.

be applied to many classification problems, especially those with a small number of classes. Consequently, as Culotta et al. [2006] rightfully pointed out, in order for the system to take full advantage of the user's feedback, the output of the PR problem must have an underlying *structure*. This way, the user only needs to give partial feedback for the system to provide a useful solution by leveraging structural dependencies. For this reason, this thesis is entitled *multimodal interactive structured prediction* instead of *multimodal interactive pattern recognition* to state that difference explicitly.

1.1.3 Multimodal interaction

In the beginning of HCI, the interaction with the computer was performed with keyboard and mouse [Card et al., 1980]. These interaction devices provide the convenience that the interpretation of the actions is deterministic. However, they can arguably be less natural for humans to communicate their intentions. On the other hand, speech, handwriting, touch, or gaze[3] are much more natural *modalities* of interaction for human beings. The downside of these modalities is that they are ambiguous by nature. Then, it is necessary a PR system to decipher the intentions of the user, e.g., an *automatic speech recognition* (ASR) is necessary to decode speech commands or user's dictations. As we have previously explained, PR systems are prone to errors. If the system fails to recognize user intentions, then the user may try to repeat the interaction. This may lead to error spirals [Oviatt and VanGent, 1996], in which the user tries on with the same modality until finally desists in favor of a more deterministic way of interaction. Also, the user will probably avoid using the non-deterministic modality in future interactions. For instance, Shilman et al. [2006] presented an interactive application where handwriting could be used to correct a text on a tactile tablet. They mentioned that in an unpublished internal survey, "many expert users quickly started skipping the correction mechanisms that entailed rewriting and could lead to ambiguous corrections. Instead they resorted to retyping the entire word using the more certain, but less efficient, soft keyboard". Hence, obtaining a high accuracy on these ambiguous modalities is key to user acceptability. Actually, modalities with error rates up to 3% can be considered acceptable, whereas error rates below 1% are considered to be excellent [LaLomia, 1994]. However, users may accept error rates between 5% and 20% if "there is a substantial payoff in terms of achieving task goals" [Frankish et al., 1995].

Hence, we know that there are different modalities that allow a more natural interaction, but what do we understand as *multimodal* interaction? Jacko and Sears [2007] defined multimodal systems in the following way:

[3] The gaze, although it is not usually used consciously to communicate a message, can be also used to obtain practical interaction information with an *eye-tracker* device. In addition, explicitly gaze interaction can be also interesting for people with physical disabilities.

> *Multimodal systems process two or more combined user input modes – such as speech, pen, touch, manual gestures, gaze, and head and body movements – in a coordinated manner with multimedia system output.*

It is broadly accepted that multimodality is the *combination* of input modalities. Nonetheless, when interacting with PR systems to correct the output, it is more natural to think of *alternative* modalities rather than *simultaneous*. That is, user interfaces may provide several interaction modalities but the user interacts with just one at a time. As Oviatt and VanGent [1996] observed, when allowing speech and pen modalities to correct text, only 0.7% of the corrections were simultaneously spoken and written. Then, there is no evidence that users make use of redundancy of input to clarify error resolution. Withal, our view of multimodality is closer to [Suhm et al., 2001]: the challenge of multimodal interfaces is to face the *combination* of the context of the application to improve the accuracy of the PR system for that modality. It is important to mention that, frequently, this challenge is similar to that of combining *simultaneous* modalities. For instance, when correcting the output of a speech recognizer with handwriting, the problem that emerges is similar to that of combining speech and pen user input as in the definition of Jacko and Sears [2007].

We can find in the literature several attempts at developing multimodal interfaces in this sense. Particularly, in the nineties there was a huge interest in using pen and spoken interaction to correct texts [Cohen et al., 1998; Frankish et al., 1995; Huerst et al., 1998; LaLomia, 1994; Oviatt and VanGent, 1996]. Nevertheless, in those works, PR systems were used as a black box so they did not take advantage of contextual information; neither they used the user's feedback to refine the system output further than the correction given by the user. It was Suhm et al. [2001] who proposed a multimodal dictation system that leveraged contextual information from the task. In that work, the user could correct speech recognition mistakes by respeaking, spelling or handwriting. Suhm et al. [2001] used pre-context and post-context information from the word being corrected and also added a bias towards frequently misrecognized words. Pre-context influence in accuracy was statistically significant, whereas post-context was not. The explanation for that was that post-context was frequently incorrect since users did not "select maximally contiguous regions of errors". On the other hand, the bias showed significant improvements in handwriting and spelling, but not in respeaking. Finally, Shilman et al. [2006] described a user interface where handwriting and pen gestures were used to as a feedback for a smart auto-completion capability. Thus, user interaction was not only used to amend the proposed correction but other mistakes in the text as well.

Although earlier work was carried out in [Brousseau et al., 1995] to enable dictation in MT, to our knowledge, the first study of a speech-enabled interactive system was Vidal et al. [2006]. In that work, an interactive MT system was described following our definition of *multimodal* and *interactive* at the same time.

Then, several scenarios were proposed where the user was expected to speak aloud parts of the current hypothesis and possibly one or more corrections. The technique consisted on rescoring the language model with the probabilities of word-based translation models [Brown et al., 1993]. Latter, several methods were proposed in [Khadivi and Ney, 2008] that took advantage of context information. The output from speech and translation was combined in a late-fusion approach, to improve the recognition of the user's feedback. More recently, Toselli et al. [2010] explored the use of the e-pen for interactive transcription of text images. In that work, the authors took advantage of the erroneously predicted word and the previous one to improve the robustness of the on-line *handwritten text recognition* (HTR) system. In addition, the feedback was used to improve the prediction of subsequent words.

Finally, eye-tracking can also be used as an input modality. For instance, [Hyrskykari et al., 2003] used gaze fixations to pop up translations when the user had difficulties to read a text in another language. More recently, the 'Speech & Eye-Tracking Enabled CAT' (SEECAT)[4] workshop aims at providing *sight translation*, where the user speaks the translation aloud instead of typing it. The major difference with previous attempts is that SEECAT intends to use the eye-tracking information in a combined way with the speech signal to improve the final speech recognition accuracy. Although the eye-tracking modality is an interesting way of acquiring implicit information from user's intention, the technology is not quite widespread and cheap enough to be considered a real alternative at this moment. Nonetheless, we hope that in a near future cheaper eye-tracking technologies emerge, such as economic eye-tracking devices or webcam based eye-trackers [Skovsgaard et al., 2011], that will allow us to build interesting interaction techniques.

1.2 Decision making in PR

In this section, we will review the *decision theory* framework as a brief reminder/introduction to the decision making problem. Decision making in PR is related to how to assign labels for a given input. In a probabilistic approach, this problem is well understood by the *Bayes decision theory* [Bishop et al., 2006; Duda et al., 2001], which aims at finding a decision rule with *minimum classification error* (MCE)[5]. In order to introduce the basics of MCE, we will give a glimpse to the decision problem in classification, where MCE can be seen in its simplest form. Basically, classification consists in, given an object or input $\boldsymbol{x} \in \mathcal{X}$, assigning a class $c \in \mathcal{C}$ to $\boldsymbol{x}$ from a finite (and typically small) set of classes $\mathcal{C}$. We measure the loss or cost of each misclassification with the

[4] http://bridge.cbs.dk/platform/?q=SEECAT
[5] Also known as *minimum Bayes risk* (MBR).

loss function[6] $\lambda(c, c^*)$, being c^* the real class[7] of $\boldsymbol{x}$. Hence, we would like to classify $\boldsymbol{x}$ with the *minimum conditional risk* (also known as expected loss) with respect to a probability distribution $\mathrm{R}_{\Pr(c|\boldsymbol{x})}(c \mid \boldsymbol{x})$,

$$\hat{c} = \arg\min_{c \in \mathcal{C}} \mathrm{R}_{\Pr(c|\boldsymbol{x})}(c \mid \boldsymbol{x}) \tag{1.1}$$

For the sake of clarity, we will omit the $\Pr(c \mid \boldsymbol{x})$ subscript in the future. In addition, the conditional risk can be also expressed as

$$\mathrm{R}(c \mid \boldsymbol{x}) = \mathrm{E}\left(\lambda(c, c') \mid \boldsymbol{x}\right) = \sum_{c' \in \mathcal{C}} \lambda(c, c') \Pr(c' \mid \boldsymbol{x}) \tag{1.2}$$

For classification tasks, λ is the so called *zero-one loss* function, $z(c, c^*) = [c \neq c^*]$, that is 1 if $c \neq c^*$ and 0 otherwise. In other words, we could say that a misclassification costs 1 unit of effort to correct the outcome or it represents 1 unit of loss (i.e., financial loss). Then, by applying the *zero-one loss* function to Equation (1.1),

$$\begin{aligned}
\hat{c} &= \arg\min_{c \in \mathcal{C}} \mathrm{R}(c \mid \boldsymbol{x}) && (1.3)\\
&= \arg\min_{c \in \mathcal{C}} \sum_{c' \in \mathcal{C}} [c \neq c'] \Pr(c' \mid \boldsymbol{x}) && (1.4)\\
&= \arg\min_{c \in \mathcal{C}} 1 - \Pr(c \mid \boldsymbol{x}) && (1.5)\\
&= \arg\max_{c \in \mathcal{C}} \Pr(c \mid \boldsymbol{x}) && (1.6)
\end{aligned}$$

which results in the well known *maximum-a-posteriori* (MAP) decision rule used by default in most classification problems.

1.3 Structured prediction

Although classification is a very useful approach, it is not appropriate to problems where the output cannot be expressed in a finite (and reduced) number of classes. If the output consists of a set of variables correlated by some structure, then the direct modeling of the class posterior probability becomes impractical, since the number of output classes can be exponential. Therefore, it is necessary to apply some kind of search algorithm over the hypothesis space. These problems are known as *structured prediction* (SP)[8] [Parker et al., 2009; Taskar et al., 2005], in which the output consists of structured labels (as opposed to independent labels in classification). Examples of structured outputs are natural

[6] The loss function can be also defined as conditioned on $\boldsymbol{x}$ [Ferrer, 2010]. However, that is usually not the case in a typical loss function. Therefore, we will omit $\boldsymbol{x}$ for the rest of the document.

[7] True state of nature [Duda et al., 2001].

[8] This was also known as syntactic or structural pattern recognition [Fu, 1982].

language sentences, DNA sequences or an XML describing the layout analysis of a web page.

In SP we can reformulate the classification problem to take into account input and output structured objects. Thus, $\boldsymbol{x} \in \mathcal{X}^*$ becomes $\boldsymbol{x} = \{x_1, x_2, \ldots, x_{|\boldsymbol{x}|}\}$, a sequence[9] of elements of $\mathcal{X}$, with length $|\boldsymbol{x}|$. Note that $\boldsymbol{x}$ belongs now to the Kleene closure of $\mathcal{X}$, that is, $\boldsymbol{x}$ has zero or more elements. Also, there is a, probably unknown, correlation among the elements in $\boldsymbol{x}$ that establishes the structure of $\boldsymbol{x}$. Then, the problem consists in obtaining a structured object $\boldsymbol{y} = \{y_1, y_2, \ldots, y_{|\boldsymbol{y}|}\}$ from an output hypothesis space $\mathcal{Y}^*$. The elements in $\boldsymbol{y}$ are correlated among them, but they are also correlated with the elements in $\boldsymbol{x}$. If we want to obtain a $\boldsymbol{y}$ that is represented by $\boldsymbol{x}$, then we can apply the MAP decision rule in the following way:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y} \in \mathcal{Y}^*} \Pr(\boldsymbol{y} \mid \boldsymbol{x}) \tag{1.7}$$

In many cases, this probability can be difficult to obtain directly and reliably. Then, it can be useful to apply the noisy-channel approach instead:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y} \in \mathcal{Y}^*} \frac{\Pr(\boldsymbol{x} \mid \boldsymbol{y}) \Pr(\boldsymbol{y})}{\Pr(\boldsymbol{x})} \tag{1.8}$$

$$= \arg\max_{\boldsymbol{y} \in \mathcal{Y}^*} \Pr(\boldsymbol{x} \mid \boldsymbol{y}) \Pr(\boldsymbol{y}) \tag{1.9}$$

where the structure of $\boldsymbol{y}$ can be modeled more easily since now it is expressed in an explicit form by $\Pr(\boldsymbol{y})$. On the other hand, the relationship between the elements in $\boldsymbol{x}$ and $\boldsymbol{y}$ is explained by the likelihood probability instead of the posterior probability.

1.3.1 Correcting the output of a SP system

Equation (1.7) and Equation (1.9) are the fundamental equations for statistical SP. They provide a means to obtain an output that is most likely to be the representation of the input. However, as we have seen in the preliminaries, SP systems are not perfect. Then, when the quality of the output is critical, it is often necessary to make a human operator correct the predicted output structure so as to meet the quality standards. We can identify three alternatives to how the output of SP systems can be enhanced.

[9] There exists other representations for structured objects that are not sequences, e.g., trees. However, as the problems we will deal with in this thesis can all be represented as sequences, we will assume this structure for simplicity.

Post-editing. Traditionally, SP has been approached as a fully automated procedure. The automatic SP scenario, depicted in Figure 1.2, can be described as follows: an input ($\boldsymbol{x}$) is presented to a SP system. Then the SP system produces an output ($\hat{\boldsymbol{y}}$), which will typically contains some errors. Finally, a human expert revises the system output. The purpose of the expert is to amend the errors to produce the final output (or reference), $\boldsymbol{r}$. The corrections are done without additional computer aid[10], e.g., by using a conventional text editor. From here on, we will refer to this process as *post-editing* (PE).

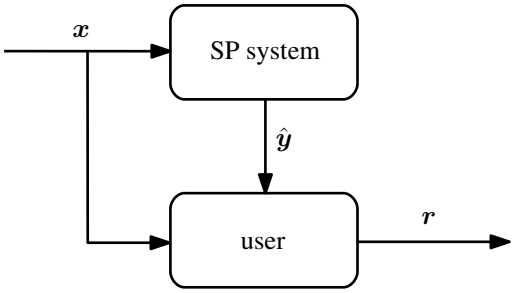


Figure 1.2: Diagram of the *post-editing* process. The system processes the input $\boldsymbol{x}$ to produce an output $\hat{\boldsymbol{y}}$. Then, the user, who knows how to obtain the desired output given $\boldsymbol{x}$, modifies $\hat{\boldsymbol{y}}$ to create the final output $\boldsymbol{r}$.

Passive interaction. The *interactive structured prediction* (ISP)[11] framework was introduced in [Vidal et al., 2007] to alleviate the cost of correcting an automatically generated output, as a generalization of the TransType2 findings. In ISP, the user is introduced in the core of a SP system so that the system and the user interact with each other to minimize the effort in producing a satisfactory output. ISP is considered passive interaction since it is the user who takes the initiative and the system behaves in a reactive (passive) way. Figure 1.3 represents the ISP interaction scheme. An input $\boldsymbol{x}$ is given to the system, which outputs a possible hypothesis $\hat{\boldsymbol{y}}$. Then, the user analyzes this output and provides feedback $\boldsymbol{f}$ regarding some of the errors committed. Now, the system can benefit from the feedback to propose a new improved hypothesis. This process is repeated until the user finds a satisfactory solution, $\boldsymbol{r}$, and the process ends. Note the loop in Figure 1.3. It indicates that for each user interaction the system has to output a new hypothesis (though it may coincide partially or totally with the previous one), and that several interactions can be performed until a satisfactory solution is found.

Active interaction. In passive interaction, the user takes the initiative when supervising the system output. Although that approach may help to reduce

[10] The user could use the computer to help her task, but the computer does not react intelligently the user's actions.

[11] Also known as *interactive pattern recognition* [Toselli et al., 2011] and closely related to *corrective feedback* [Culotta et al., 2006].

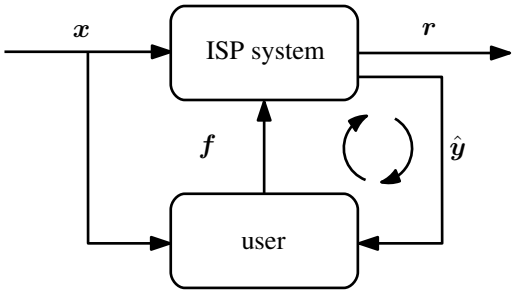


Figure 1.3: Diagram of an *passive interactive structured prediction* process. The system processes the input $\boldsymbol{x}$ to produce an initial output $\hat{\boldsymbol{y}}$. Then, the user analyses the output and proposes a correction by some feedback $\boldsymbol{f}$. Now, the system proposes a new hypothesis $\hat{\boldsymbol{y}}$. This process is repeated until the desired solution $\boldsymbol{r}$ is obtained.

the human effort, the user still has to supervise the entire collection of system outputs. This can be a waste of effort, especially for the outputs that are very likely to be correct. In this context, active interaction [Oncina and Vidal, 2011; Toselli et al., 2011] can be very beneficial. As we mentioned in the introduction, active interaction is close in concept to active learning [Settles, 2010] in the sense that the system takes the initiative to propose a sample for the user to correct or annotate. However, in contrast to active learning, the goal of active interaction is to minimize the effort in obtaining an error-free (or a certain degree of error) output. Figure 1.4 is a representation of a typical active system. In each iteration, the system outputs a hypothesis and asks the user to supervise a particular element of the output. Then, the user accepts if the label is correct or rejects it, providing in this case the correct label.

In these three scenarios, we would like to find a decision rule that allows us to obtain the final solution with less user effort. Hence, the *zero-one* loss function used to obtain Equation (1.7) and Equation (1.9) may not longer be optimal.

1.4 Decision making in post-editing

It is well known that the MAP decision rule optimizes the zero-one loss function, which assign 1 to an incorrect output regardless of how many errors have been produced, i.e., an output that fails to predict one element is considered just as wrong as a sentence that mispredicts each and every element. The zero-one loss function may be still of interest for many SP tasks. For instance, an automatic postal code classifier puts letters on different boxes depending on their postal code. It has the same cost associated to outcomes that have one or more mistakes, since they all go to the wrong box. On the other hand, in the recognition of a text book, we would like to have the less character errors as possible, and thus, the zero-one loss function may not be so appropriate in this case.

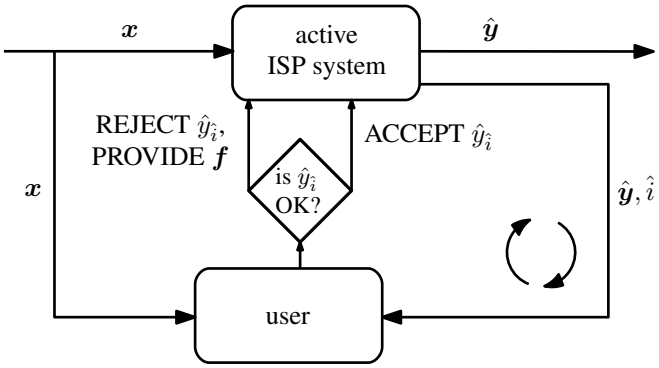


Figure 1.4: Diagram of an *active interactive structured prediction* process. The system processes the input $\boldsymbol{x}$ to produce an initial output $\hat{\boldsymbol{y}}$. Then, the system selects an element of the output structure in position i, ($\hat{y}_i$), and asks the user to correct it. The user analyzes the query and, in case there is an error, proposes a correction with some feedback $\boldsymbol{f}$. Now, the system updates the hypothesis $\hat{\boldsymbol{y}}$ given user's feedback and asks for a new correction. This process is repeated until the system decides that there should be no more errors; or if the user has surpassed a given quota of interactions. Thus, the final result might be different from the expected result $\boldsymbol{r}$.

There are several means to define a loss function for SP problems. Most of them measure the accuracy or error at element level. However, in this thesis we are interested in measuring the PE effort, i.e., the effort that is needed to fix the erroneous output elements to obtain the correct or expected output. For our purposes, we can differentiate two kinds of SP problems: sequence labeling problems and problems with an unknown number of output elements.

1.4.1 Sequence labeling

The sequence labeling problem is characterized by the fact that the output object $\boldsymbol{y}$ has the same length than the input object $\boldsymbol{x}$. Besides, the correspondence between the elements in $\boldsymbol{x}$ and $\boldsymbol{y}$ is one-to-one and, frequently, they are monotonically related, i.e., x_n is related to y_n for each n. Problems that fall under this category are *optical character recognition* (OCR), *part-of-speech tagging* and the *assignment problem*. The post-editing loss function in this case can be modeled with the Hamming distance [Hamming, 1950] that computes the number of positions at which the corresponding symbols are different, i.e., $h(\boldsymbol{y}, \boldsymbol{y}^*) = \sum_{i=1}^{|\boldsymbol{y}|} [y_i \neq y_i^*]$. In terms of PE, the Hamming distance computes the number of *substitutions* that are needed to obtain the correct solution. Therefore, an optimum decision rule can be obtained by minimizing the conditional risk in Equation (1.1) for the Hamming distance:

$$\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y}\in\mathcal{Y}^*} \sum_{\boldsymbol{y}'\in\mathcal{Y}^*} h(\boldsymbol{y},\boldsymbol{y}') \Pr(\boldsymbol{y}' \mid \boldsymbol{x}) \tag{1.10}$$

$$= \arg\min_{\boldsymbol{y}\in\mathcal{Y}^*} \sum_{\boldsymbol{y}'\in\mathcal{Y}^*} \sum_{i=1}^{|\boldsymbol{x}|} [y_i \neq y'_i] \Pr(\boldsymbol{y}' \mid \boldsymbol{x}) \tag{1.11}$$

$$= \arg\min_{\boldsymbol{y}\in\mathcal{Y}^*} \sum_{i=1}^{|\boldsymbol{x}|} \left[\sum_{\boldsymbol{y}'\in\mathcal{Y}^*} [y_i \neq y'_i] \Pr(\boldsymbol{y}' \mid \boldsymbol{x}) \right] \tag{1.12}$$

Then, minimizing the risk for each position is a sufficient condition to minimize the sample risk:

$$\hat{y}_i = \arg\min_{y_i\in\mathcal{Y}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^*} [y_i \neq y'_i] \Pr(\boldsymbol{y}' \mid \boldsymbol{x}) \tag{1.13}$$

$$= \arg\max_{y_i\in\mathcal{Y}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^*:y_i=y'_i} \Pr(\boldsymbol{y}' \mid \boldsymbol{x}) \tag{1.14}$$

$$= \arg\max_{y_i\in\mathcal{Y}} \Pr(y_i \mid i, \boldsymbol{x}) \tag{1.15}$$

The last equation is the so-called *position posterior probability* that usually can be computed efficiently using a *forward-backward*-like algorithm [Chelba and Acero, 2005]. It is important to note that a decision rule that follows MCE for a specific cost function, does not necessarily always obtain the best result, but the best result in *average*. In addition, MCE decision rules may generate solutions that are not correct with respect to the structure constraints. For instance, when performing English OCR with Equation (1.15), the output generated by the decision rule could be a sequence of characters that does not form a word in English. Figure 1.5 illustrates this problem. In this case, $\boldsymbol{x}$ is a sequence of handwritten characters representing the word 'MONK'. The three top hypothesis $(\boldsymbol{y}', \boldsymbol{y}'', \boldsymbol{y}''')$ along their respective posterior probabilities are shown below. At the bottom, there is the result given by Equation (1.15). We can see that hypotheses $\boldsymbol{y}''$ and $\boldsymbol{y}'''$ agree in position 1. As a result, the position posterior probability of 'M' in position 1 is higher than that of 'H'. The resulting word is not an English word. Whereas the MAP solution, $\boldsymbol{y}'$, has two errors, $\hat{\boldsymbol{y}}$ has only one, and thus the PE effort is lower. The rationale behind it is that by forcing an output with a correct structure more errors can be introduced. However, by obtaining the best element at each position we ensure that the number of corrections to be made are minimum, although the structure of the output is not correct. And that is precisely the goal we pursuit in MCE for PE.

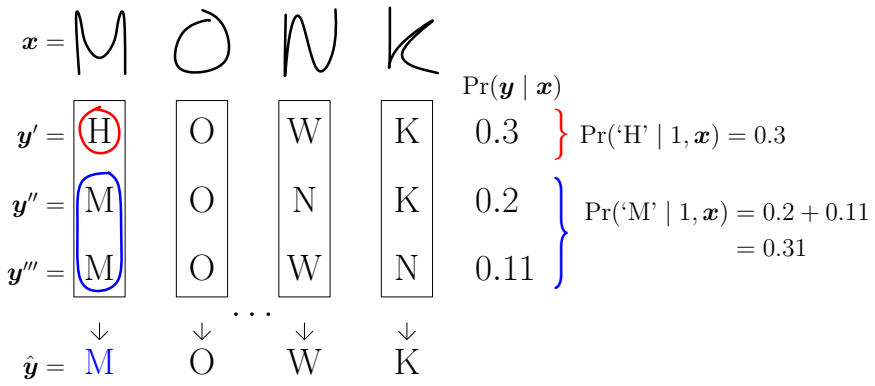


Figure 1.5: OCR example for the handwritten word 'MONK'. The hypothesis given by a MAP approach has two errors. However, the MCE approach, which accumulates the probability of several hypotheses, has only one error. Note that the MCE hypothesis is not an English word but needs less corrections.

1.4.2 Unknown alignment and output size

In the aforementioned OCR problem, we know that the output character y_i is produced as the transcription of character image x_i. Thus, we say that y_i is aligned to x_i since x_i is the responsible for producing y_i, or vice versa, y_i explains x_i. Accordingly, the alignment of the input/output elements in sequence labeling is, by definition, established by its position. In contrast, there is a kind of SP problems in which the alignment is unknown. In such cases, the position of an output element does not determine what input elements produced it. Often, such correlation needs to be expressed and modeled in the form of hidden (unobserved) variables that define these alignments. In addition, these problems usually present the difficulty that the output size is also unknown. Hence, we also must take into account the output hypothesis being shorter or longer than the input. Therefore, the Hamming distance cannot be applied as a function to measure the PE effort. Consequently, two additional operations must be introduced: *deletion* of elements that are not in the correct solution, and *insertion* of elements that are not in the system hypothesis. These operations, together with the *substitution* operation in the Hamming distance conform a basic set of edit operations[12]. Note that, if the edit distance is properly normalized by the number of words in the reference, the resulting function is the *word error rate* (WER) function that is used to evaluate many natural language problems. In consequence, a loss function to estimate PE human effort can be obtained by the minimum number of edit

[12] The Hamming distance can be seen as a particular case of the edit distance where only substitutions are allowed. In fact, Schlüter et al. [2010] established that the Hamming risk is an upper bound to the edit distance risk, which in practice leads to an algorithm that is locally optimal w.r.t. the edit distance risk [Stolcke et al., 2000].

operations necessary to convert the system output into the correct solution. This can be obtained with the edit distance [Levenshtein, 1966], $e(\boldsymbol{y}, \boldsymbol{y}*)$. Note that this estimation of the PE effort is optimistic, since a user is not likely to perform the actual *minimum* number of operations. Instead, she will try to spend less time doing the task, but with no guaranties of optimality. Taking that into consideration, the optimal rule can be obtained by:

$$\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y} \in \mathcal{Y}} \sum_{\boldsymbol{y}' \in \mathcal{Y}} e(\boldsymbol{y}, \boldsymbol{y}') \Pr(\boldsymbol{y}' \mid \boldsymbol{x}) \tag{1.16}$$

This equation cannot be decomposed into smaller terms, and there are no known efficient exact solutions to it. However, approximate decoding algorithms exist which are based on *confusion networks* (CN) [Mangu et al., 1999] and lattice segmentation [Goel et al., 2004]. Note that these decoding strategies favor solutions with less errors and, hence, they are cheaper to amend. Instead, the *zero-one* loss function does not distinguish between a solution with just one error or a completely wrong solution. Unfortunately, [Schlüter et al., 2012] analytical results conclude that, for integer-valued metric cost functions, the MAP class often dominates the Bayes decision rule. As a result, limited improvements can be expected from using the optimal decision rule, especially for tasks with low error rate. Nevertheless, the study of optimal decision rules is still an interesting and challenging problem that provides insight of the task at hand.

1.5 Passive interaction

In the SP community, the majority of attempts to improve the quality of the output have been focused on developing systems that produce less "errors", for instance by improving the decision rule as has been shown in the previous sections. In that sense, the scheme in Figure 1.2 has been implicitly adopted: a system with less "errors" would allow a manual PE of the system output with less effort. However, PE delegates all the effort to the user. So, what can we do to reduce the effort needed to correct the output?

Consider the PE of the SP example in Figure 1.6 which depicts an OCR problem of handwritten text in a form. The structure of this form is the following: the four first fields are digits, while the fifth field is an error-detecting code. This code is an ASCII character which position in the ASCII table can be computed as the sum of the numbers of the four first fields modulo 26 plus 65. This condition poses a strong restriction on the structure of the output. In this case of Figure 1.6, the real transcription of the image is $3527R$ and the last field is computed as $chr(65+3+5+2+7) =$'R'. Note that the third field is a '3' with a superimposed bold '2' which can be easily confused by a '8'. In addition, the fourth field can be confused with a '1'. Let us assume that 'R' is correctly

recognized. Hence, a system could mistakenly recognize the whole sequence as $3581R$, since the error-detecting code is also 'R'.

Figure 1.6: Example of handwritten text on a form. The transcription is $3527R$. The last field is an error-detecting code that can be computed as the ASCII character at the position 65 plus the numbers in the previous fields ($chr(65+3+5+2+7)$ ='R').

In such example where the system has output $3581R$, would PE be an optimal strategy to fix the system's output? PE takes two actions to correct the output. First, '8' can be substituted by '2', and then, '1' by '7'. Note the use of the Hamming distance since this is a sequence labeling PE problem. However, given that the system may have some knowledge of the underlying structure of the output, this process can be performed more efficiently. Suppose that the user substitutes '8' by '2'. Suppose as well that the system has strong evidence (i.e., the position posterior probability is one) that the first, second and last fields are correct, i.e. the system believes that 3, 5, and R are the correct labels. Then, it is most likely that the fourth field is a '7', and not '1', so that the error-detecting code adds up. Hence, the system can automatically replace '1' by '7' in the fourth field, obtaining the correct solution with just one substitution. In this simplified scenario, such a system would reduce the user effort in correcting the initial output. This process can be defined as interactive since the user and the system have collaborated to produce the final output. On the one hand, the user participates by controlling the generation of the output, i.e., the user amends parts of the output. On the other hand, the system takes user's amendments into consideration to predict a new improved solution.

1.5.1 Sequential passive interaction

A particular instance of ISP is when the user corrects the elements in the output in a left-to-right fashion. Thus, $\boldsymbol{y}$ can be split into a prefix, $\boldsymbol{y}_p$, that the user has validated (and therefore is correct), and suffix, $\boldsymbol{y}_s$, that may contain some errors. Consequently, the output is the concatenation of the prefix and the suffix, $\boldsymbol{y} = \boldsymbol{y}_p \cdot \boldsymbol{y}_s$. With the superscript $^{(i)}$ we will indicate that the variable was produced in the iteration number i between the user and the system. Thus, in each user interaction, the user gives a feedback as a substitution of the first erroneous word in the suffix by the correct word in position k (of the reference $\boldsymbol{r}$), $\boldsymbol{f}^{(i)} = r_k$. The validated prefix and the user feedback are concatenated to form the new prefix $\boldsymbol{y}_p^{(i)}$. Then, the ISP system uses a decision rule to obtain a new suffix in which the correction introduced by the user is taken into account.

Typically, ISP systems have used an extension of the MAP decision rule. The underlying loss function for this decision rule assigns a loss of 1 if the suffix

has one or more errors and 0 if the suffix is completely correct. Thus, at each interaction (i) a new hypothesis is obtained conditioned on the feedbacks from previous interactions, which are encoded in $\boldsymbol{y}_p$ as part of the validated and corrected prefix:

$$\hat{\boldsymbol{y}}_s^{(i)} = \arg\max_{\boldsymbol{y}_s} \Pr(\boldsymbol{y}_s \mid \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \tag{1.17}$$

The MAP decision rule has been successfully applied to several SP tasks [Toselli et al., 2011], namely interactive machine translation [Barrachina et al., 2009], interactive transcription of text images [Toselli et al., 2010], interactive predictive parsing [Sánchez-Sáez et al., 2009], interactive speech transcription [Rodríguez et al., 2007] and interactive text generation [Rodríguez et al., 2010]. However, the effort of correcting an output in ISP is measured as the number of corrections (or interactions) needed to obtain the correct solution. As it was the case in PE, we will see in Chapter 3 why the MAP decision rule is not optimal to minimize the number of corrections, and how to derive the MCE decision rule for sequential ISP.

1.6 Active interaction

While passive interaction can be helpful in producing the correct output, it is also worth noticing that the user is expected to supervise the whole system output. This does not seem a big problem when dealing with the example in Figure 1.6, but it can be a real drawback when transcribing the national identification numbers of several thousands of handwritten forms. In the latter case, there might be just a few errors located unevenly throughout the form transcriptions. Then, it seems natural to let the system decide what should be supervised, on the basis of the confidence the system has regarding the correctness of the solution. Ideally, that way would allow the allocation human resources only on the labels that need correction, but not on the solutions that can be deemed as acceptable.

As we have seen in Figure 1.4, in active learning we aim at finding a decision rule that can choose output labels for the user to supervise in a way that the effort is minimized. Thus, we can say that, after each user feedback, we obtain the acceptance of the system's proposal or a correction of it, as the position and correct label. Let $\boldsymbol{h}^{(i-1)} = \{(k^{(1)}, y^{(1)}), (k^{(2)}, y^{(2)}), \ldots, (k^{(i-1)}, y^{(i-1)})\}$ be the history of positions and corrections given by the user up to iteration (i). A decision rule, $\mathcal{S}$, that chooses the next output label to supervise at iteration (i) should comply to the following expression:

$$(\hat{k}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) = \mathcal{S}(\boldsymbol{x}, \boldsymbol{h}^{(i-1)}) \tag{1.18}$$

where $\hat{k}^{(i)}$ is an output position in the hypothesis $\hat{\boldsymbol{y}}^{(i)}$. Expressed in a more compact way, $\mathcal{S}$ decides which $\hat{y}_{\hat{k}}^{(i)}$ the user should correct conditioned to the

input and the history of user feedbacks. Active interaction can be also be performed at structure level. In this case, the system goal is to retrieve structures or objects from a pool in such a way that already correct or almost correct structures are left unsupervised.[13] In PE and ISP, the selection of the loss function, although not ideal, was quite a natural choice. However, in active interaction, we can define at least two loss functions depending on the nature of the problem.

Reducing the number of supervisions. Here, we assume that supervision has a high cost, even if the output is correct. For instance, in the case of having a thousand handwritten forms, the whole collection should be supervised in order to achieve an overall high quality result. The supervision of a form that is already correct has, indeed, a high cost. In fact, the human operator needs at least to solve the problem mentally to check if the output is correct. Thus, it can be very resource consuming to check all the forms. That is especially true for the kind of systems that can achieve very good accuracy. In those cases, the error is concentrated in the output of some specific inputs that are much harder to decode. Then, it would be very helpful to have a system that only queries the user for outputs containing errors. This kind of active interaction can be understood as a *quality estimation* problem, as it is approached in the machine translation community [Callison-Burch et al., 2012].

Minimizing the number of corrections. In this second approach, the goal is to query the user for specific output labels that need to be accepted or corrected in order to minimize the user effort when correcting labels, but not necessarily reduce the number of supervisions. Thus, we assume that the effort in correcting a label is much higher than the effort in accepting a correct label. This may seem not to hold true for the example in Figure 1.6, since the effort in typing the correct character with a keyboard is the same as the effort needed to accept the correct label, a single keypress. In addition, the cognitive effort to decide if the label of a tainted handwritten character is correct or not, also gives as a by-product the correct label. Thus, in this case the assumption would not be true. Nevertheless, there are cases where the effort in correcting the output can be higher, e.g., correct complete words in handwritten text recognition. Examples of a kind of interaction scenario similar to this one are [Serrano et al., 2010] and [González-Rubio et al., 2010], for handwritten text recognition and machine translation respectively. However, those works did not leveraged the user feedback to propagate the corrections so that the suggestion in the next iteration could be improved.

Although both loss functions may seem much alike, there is a major difference that can make them incompatible. In the second case, supervising a correct label has negligible cost. Thus, the decision rule may decide to query the user

[13] In a streaming scenario, where structures are being constantly received and they cannot or should not be stored, an active interactive system indicates if a received structure should be amended or passed on unsupervised to the next stage.

to supervise many correct labels. However, this would be against the goals of the first approach. In Chapter 4, we will analyze both loss functions and, for each of them, we will compare a series of decision rules.

1.7 Multimodal interaction

Typically, the way the feedback is introduced to the system is by means of the keyboard or, in a more advanced scenario, implicit mouse actions [Sanchis-Trilles et al., 2008]. Additionally, other feedback modalities can be found to be more productive, as speech [Dragsted et al., 2011], or more ergonomics, as handwriting [Toselli et al., 2010] especially in tactile devices. However, in the former the feedback can be determined in a deterministic manner, whereas the latter are non deterministic modalities. As such, a SP system must be build to interpret the feedback signal, which can result in erroneous interpretations. If the feedback in iteration (i), $\boldsymbol{f}^{(i)}$, is decoded into a sequence of elements, $\boldsymbol{d}^{(i)}$, by a completely decoupled black-box system, then we can apply the following decision rule:

$$\widehat{\boldsymbol{d}}^{(i)} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}^{(i)}) \tag{1.19}$$

Non-determinism poses a problem to these modalities with respect to the deterministic ones. Since the interpretation may be wrong the use of the modality might be hindered [Shilman et al., 2006]. Hence, it is necessary to make such systems more robust. We can make this by two different means. First, we can make Equation (1.19) context-aware so that the input, $\boldsymbol{x}$, and all previous feedbacks, $\boldsymbol{h}^{(i)}$, are considered,

$$\widehat{\boldsymbol{d}}^{(i)} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{x}, \boldsymbol{h}^{(i)}, \boldsymbol{f}^{(i)}) \tag{1.20}$$

This way, $\boldsymbol{x}$ and $\boldsymbol{h}^{(i)}$ can be seen as constraints to the search of $\boldsymbol{x}$ given $\boldsymbol{f}^{(i)}$ in the classical maximization problem in Equation (1.19).

Second, the decoding of the feedback can be integrated directly into the system prediction. Thus, Equation (1.17) can be modified to take into account all the possible decodings $\boldsymbol{d}$ of the feedback signal:

$$\hat{\boldsymbol{y}}^{(i)} = \arg\max_{\boldsymbol{y}} \sum_{\boldsymbol{d}} \Pr(\boldsymbol{y}, \boldsymbol{d} \mid \boldsymbol{x}, \boldsymbol{h}^{(i)}, \boldsymbol{f}^{(i)}) \tag{1.21}$$

However, $\boldsymbol{d}$ must be a subsegment of $\boldsymbol{y}$ by definition, and thus, $\boldsymbol{d}$ must be consistent with $\boldsymbol{y}$. Furthermore, in the scenarios that we will study, the position of $\boldsymbol{d}$ in $\boldsymbol{y}$ is known. Then, if $\boldsymbol{d}$ should be placed in position k, we are restricted

to outputs where $d_1^{|d|} = y_k^{k+|d|}$. Hence, Equation (1.21) can be rewritten as

$$\hat{\boldsymbol{y}}^{(i)} = \underset{\boldsymbol{y}: d_1^{|d|} = y_k^{k+|d|}}{\arg\max} \Pr(\boldsymbol{y}, \boldsymbol{d} \mid \boldsymbol{x}, \boldsymbol{h}^{(i)}, \boldsymbol{f}^{(i)}) \tag{1.22}$$

since only one $\boldsymbol{d}$ can be consistent with $\boldsymbol{y}$.

In Chapter 5 and Chapter 6, we will explain in more detail how these strategies can be done for HTR and ASR, respectively.

1.8 Objectives of the thesis

The objective of this thesis is two folded: on the one hand, it aims to provide a theoretical and empirical study of decision rules for interactive structured prediction; on the other hand, this thesis aims to find algorithms that can integrate non-deterministic input modalities, such as speech or on-line handwriting, in a robust and efficient manner. Finally, the design and evaluation of a working prototype is described. More precisely, the scientific contributions to this thesis and the resulting publications are:

1. **Optimum decision rule for passive interaction.** [**Alabau et al., 2012c**] Traditionally, the systems that require PE follow a strategy to minimize output errors. However, this strategy is not optimum for ISP since the strategy should be formalized in terms of minimizing user interactions. To this respect, this work aims to give insight into the optimal decision rule for ISP, find efficient algorithms and asses the proposed methods with real world SP problems. Here, inspired by [Oncina, 2009; Oncina and Vidal, 2011], we have delved into an optimum decision rule for ISP which covers a broader range of common ISP problems, and where the output depends on a structured input. We analyze the strategy from a theoretical perspective and also develop a practical decoding algorithm that can be used straightforwardly in many SP problems. In addition, we show that the traditional decision rule that has been used for ISP so far is a good approximation to the optimum for ISP, which is confirmed by the experiments.

2. **Decision rules for active interaction.** Active interaction differs from passive interaction in that it is the computer that takes the initiative and selects an object or element for the user to label. We describe active interaction as a different yet related problem to active learning, and take advantage of the taxonomy of active learning solutions to adapt them to the active interaction scenario. We analyze two active interaction scenarios. In the first one, the system asks the user to correct a specific object from a collection of objects, with the intention to avoid the supervision of correct outputs. In the second one, the system asks the user to correct an

element of an object, so as to diminish the number of interactions needed to obtain the correct solution.

3. **Interaction with an electronic pen.** [**Alabau and Casacuberta, 2012**; **Alabau et al., 2010**, **2011c**, **2013**] Currently, tactile devices are almost ubiquitous. With a screen roughly the size of an A5 paper, some of those devices seem ideal to use handwriting as a means of introducing and amending text. However, handwriting recognizers commit enough errors to hinder its use. In this thesis we aim at providing a more robust handwriting recognition by leveraging contextual information. In particular, we focus our efforts towards an interactive machine translation scenario. We combine information from the handwriting signal, the source sentence, and previous user interactions to improve recognition accuracy. Besides, we present an extended and detailed analysis of the experimentation process, which identifies that major source of recognition errors. Finally, we propose a method to recover from HTR errors, which can reduce the number of characters introduced to a quarter.

4. **Speech interaction and dictation.** [**Alabau et al., 2011a**,**b**, **2012b**] An alternative to use the keyboard and the mouse to interact with the system is by issuing commands with the voice. Interacting with speech is a difficult problem, since the speech error rates can be high. This part of the thesis is devoted to exploring new techniques for fusing the translation/transcription and speech inputs to provide a more reliable speech-enabled input interface that can lead to a real multimodal system. We take on the work from Vidal et al. [2006] in speech interaction with interactive MT systems to performing a better integration. In addition, we extend the work to speech dictation in HTR. We propose a technique, based on word graphs, that allows a context aware decoding. In MT, compared to word-based translation models, this technique exhibits an important increase in recognition performance. With respect to HTR, we compare the use of speech dictation to transcribe handwritten text documents against the direct use of text recognition.

5. **Prototype design and evaluation.** [**Alabau et al., 2012a**] Recent developments in search algorithms and software architecture have enabled multi-user web-based prototypes for interactive systems. Surprisingly, formal human evaluations of these prototypes are highly scarce in the literature. During the last years, we have developed several interactive web prototypes, which are accessible worldwide, that need to be validated in the field. To this regard, we aim to asses the web prototypes. On the one hand, we would like to know if the casual visitor notices the advantages of the interactive prototype. On the other hand, we want to leverage the evaluation to see if the perception was really represented by actual performance. Two rounds of evaluations have been performed on the interactive MT prototype, comparing it to a PE prototype (an

interactive MT one but with disabled interaction capabilities). These evaluations have given some insights on the current technology as well as discovered in which areas future research efforts should be addressed.

Bibliography

V. Alabau and F. Casacuberta. Study of Electronic Pen Commands for Interactive-Predictive Machine Translation. In *International Workshop on Expertise in Translation and Post-editing Research and Application*, 2012.

V. Alabau, D. Ortiz-Martínez, A. Sanchis, and F. Casacuberta. Multimodal Interactive Machine Translation. In *Proc. of the International Conference on Multimodal Interfaces (ICMI-MLMI'10)*, p. 46:1–46:4, 2010.

V. Alabau, L. Rodríguez-Ruiz, A. Sanchis, P. Martínez-Gómez, and F. Casacuberta. On multimodal interactive machine translation using speech recognition. In *Proc. of the International Conference on Multimodal Interaction (ICMI'11)*, p. 129–136, 2011a.

V. Alabau, V. Romero, A. L. Lagarda, and C. D. Martínez-Hinarejos. A Multimodal Approach to Dictation of Handwritten Historical Documents. In *Proc. of Interspeech'11*, p. 2245–2248, 2011b.

V. Alabau, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*, p. 389–394, 2011c.

V. Alabau, L. A. Leiva, D. Ortiz-Martínez, and F. Casacuberta. User Evaluation of Interactive Machine Translation Systems. In *Proc. of the 16th Annual Conference of the European Association for Machine Translation (EAMT'12)*, p. 20–23, 2012a.

V. Alabau, C. D. Martínez-Hinarejos, V. Romero, and A. L. Lagarda. An iterative multimodal framework for the transcription of handwritten historical documents. *Pattern Recognition Letters*, in press, 2012b.

V. Alabau, A. Sanchis, and F. Casacuberta. On the Optimal Decision Rule for Sequential Interactive Structured Prediction. *Pattern Recognition Letters*, 33(6):2226–2231, 2012c.

V. Alabau, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition in Interactive Machine Translation. *Pattern Recognition*, submitted, 2013.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J. Vilar. Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1):3–28, 2009.

C. Bishop et al. *Pattern recognition and machine learning*. Springer, 2006.

J. Brousseau, C. Drouin, G. Foster, P. Isabelle, R. Kuhn, Y. Normandin, and P. Plamondon. French speech recognition in an automatic dictation system for translators: the TransTalk project. In *Proc. of Eurospeech'95*, p. 193–196, 1995.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The Mathematics of Machine Translation. *Computational Linguistics*, 19(2):263–311, 1993.

R. Brown and S. Nirenburg. Human-computer interaction for semantic disambiguation. In *Proc. of the 13th Conference on Computational Linguistics (COLING'90)* , p. 42–47, 1990.

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proc. of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, p. 10–51, 2012.

S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Communnications of the ACM*, 23(7):396–410, 1980.

J. H. Carlisle. Evaluating the impact of office automation on top management communication. In *Proc. of the National Computer Conference and Exposition (AFIPS'76)*, p. 611–616, 1976.

C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, p. 443–450, 2005.

P. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Clow, and I. Smith. The efficiency of multimodal interaction: A case study. In *Proc. of the International Conference on Spoken Language Processing (ICSLP'98)*, volume 2, p. 249–252, 1998.

A. Culotta, T. Kristjansson, A. McCallum, and P. Viola. Corrective Feedback and Persistent Learning for Information Extraction. *Artificial Intelligence*, 170:1101–1122, 2006.

A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-computer interaction*. Prentice-Hall, 3 edition, 2004.

B. Dragsted, I. M. Mees, and I. G. Hansen. Speaking Your Translation: Students' First Encounter with Speech Recognition Technology. *The Translation & Interpreting*, 3(1): 10–43, 2011.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2. edition, 2001.

J. Esteban, J. Lorenzo, A. Valderrábanos, and G. Lapalme. Transtype2-an innovative computer-assisted translation system. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, p. 94–97, 2004.

J. A. Ferrer. *Statistical approaches for natural language modelling and monotone statistical machine translation*. PhD thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2010.

G. Foster, P. Isabelle, and P. Plamondon. Word completion: A first step toward target-text mediated IMT. In *Proc. of the 16th Conference on Computational Linguistics (COLING'96)*, p. 394–399, 1996.

C. Frankish, R. Hull, and P. Morgan. Recognition accuracy and user acceptance of pen interfaces. In *Proc. of the Conference on Human Factors in Computing Systems (CHI'95)*, p. 503–510, 1995.

K. Fu. *Syntactic pattern recognition and applications*. Prentice-Hall, 1982.

V. Goel, S. Kumar, and W. Byrne. Segmental minimum Bayes-risk decoding for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 12(3):234–249, 2004.

J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta. Balancing User Effort and Translation Error in Interactive Machine Translation Via Confidence Measures. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, p. 173–177, 2010.

L. O. Gorman. What is Pattern Recognition? *IAPR Newsletter*, 25(1):1–2, 2003.

R. Hamming. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal*, 26(2):147–160, 1950.

E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. of the Conference on Human Factors in Computing Systems (CHI'99)*, p. 159–166, 1999.

W. Huerst, J. Yang, and A. Waibel. Interactive error repair for an online handwriting interface. In *Proc. of the Conference on Human Factors in Computing Systems (CHI'98)*, p. 353–354, 1998.

A. Hyrskykari, P. Majaranta, and K. Räihä. Proactive response to eye movements. In *Proc. of INTERACT'03*, volume 3, p. 129–136, 2003.

J. A. Jacko and A. Sears, editors. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. CRC, 2007.

A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

M. Kay and G. Martins. The MIND system. *Research Memoranda*, 1970.

S. Khadivi and H. Ney. Integration of Speech Recognition and Machine Translation in Computer-Assisted Translation. *IEEE Transactions on Audio, Speech and Language Processing*, 16(8):1551–1564, 2008.

P. Koehn. A process study of computer-aided translation. *Machine translation*, 23(4): 241–263, 2009.

M. LaLomia. User acceptance of handwritten recognition accuracy. In *Proc. of the Conference on Human Factors in Computing Systems (CHI'94)*, p. 107–108, 1994.

V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics-Doklady*, 10(8):707–710, 1966.

J. Liu, J. Sun, and S. Wang. Pattern recognition: An overview. *IJCSNS International Journal of Computer Science and Network Security*, 6(6):57–61, 2006.

L. Mangu, E. Brill, and A. Stolcke. Finding Consensus Among Words: Lattice-Based Word Error Minimization. In *Proc. of Eurospeech'99*, volume 1, p. 495–498, 1999.

P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta. Online Learning via Dynamic Reranking for Computer Assisted Translation. In *Proc. of the 12th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'11)*, p. 93–105, 2011.

L. Nepveu, G. Lapalme, P. Langlais, and G. Foster. Adaptive Language and Translation Models for Interactive Machine Translation. In D. Lin and D. Wu, editors, *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*, p. 190–197, 2004.

J. Oncina. Optimum Algorithm to Minimize Human Interactions in Sequential Computer Assisted Pattern Recognition. *Pattern Recognition Letters*, 30(6):558–563, 2009.

J. Oncina and E. Vidal. Interactive Structured Output Prediction: Application to Chromosome Classification. In *Proc. of the 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'11)*, p. 256–264, 2011.

D. Ortiz-Martínez. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation*. PhD thesis, Universidad Politécnica de Valencia, 2011.

S. Oviatt and R. VanGent. Error resolution during multimodal human-computer interaction. In *Proc. of the 4th International Conference on Spoken Language Processing (ICSLP'96)*, volume 1, p. 204–207 vol.1, 1996.

C. Parker, Y. Altun, and P. Tadepalli, editors. *Machine Learning*, volume 77, 2009.

A. Ramachandran, A. Dasgupta, N. Feamster, and K. Weinberger. Spam or ham?: characterizing and detecting fraudulent "not spam" reports in web mail systems. In *Proc. of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS'11)*, p. 210–219, 2011.

L. Rodríguez, F. Casacuberta, and E. Vidal. Computer Assisted Transcription of Speech. In *Proc. of the 3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'07)*, volume 4477, p. 241–248, 2007.

L. Rodríguez, A. Revuelta, I. García-Varea, and E. Vidal. Interactive Text Generation for Information Retrieval. In *Proc. of the 10th International Workshop on Pattern Recognition in Information Systems (PRIS'10)*, p. 62–71, 2010.

L. Rodríguez-Ruiz. *Pattern Recognition Applied to Natural Language Processing.* PhD thesis, Universidad Politécnica de Valencia, 2010.

G. Sanchis-Trilles, D. Ortiz-Martínez, J. Civera, F. Casacuberta, E. Vidal, and H. Hoang. Improving Interactive Machine Translation via Mouse Actions. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, 2008.

R. Schlüter, M. Nussbaum-Thom, and H. Ney. On the relation of Bayes Risk, Word Error, and Word Posteriors in ASR. In *Proc. of Interspeech'10*, p. 230–233, 2010.

R. Schlüter, M. Nussbaum-Thom, and H. Ney. Does the Cost Function Matter in Bayes Decision Rule? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2): 292–301, 2012.

N. Serrano, A. Sanchis, and A. Juan. Balancing Error and Supervision Effort in Interactive-Predictive Handwritten Text Recognition. In *Proc. of the 15th International Conference on Intelligent User Interfaces (IUI 2010)*, p. 373–376, 2010.

B. Settles. Active learning literature survey. Tech. report, 2010.

M. Shilman, D. S. Tan, and P. Simard. CueTIP: a mixed-initiative interface for correcting handwriting errors. In *Proc. of the 19th annual ACM symposium on User interface software and technology (UIST'06)*, p. 323–332, 2006.

H. Skovsgaard, J. S. Agustin, S. A. Johansen, J. P. Hansen, and M. Tall. Evaluation of a remote webcam-based eye tracker. In *Proc. of the 1st Conference on Novel Gaze-Controlled Applications (NGCA'11)*, p. 7:1–7:4, 2011.

A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. R. Gadde, M. Plauché, C. Richey, E. Shriberg, K. Sönmez, F. Weng, and J. Zheng. The SRI March 2000 Hub-5 conversational speech transcription system. In *Proc. of the NIST Speech Transcription Workshop*, 2000.

B. Suhm, B. Myers, and A. Waibel. Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction*, 8(1):60–98, 2001.

R. Sánchez-Sáez, J. A. Sánchez, and J. M. Benedí. Interactive Predictive Parsing. In *Proc. of the 11th International Conference on Parsing Technologies (IWPT'09)*, p. 222–225, 2009.

B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: a large margin approach. In *Proc. of the 22nd International Conference on Machine Learning (ICML'05)*, p. 896–903, 2005.

A. Toselli, E. Vidal, and F. Casacuberta, editors. *Multimodal Interactive Pattern Recognition and Applications*. Springer, 2011.

A. H. Toselli, V. Romero, M. Pastor, and E. Vidal. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.

E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-Assisted Translation Using Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 14(3):941–951, 2006.

E. Vidal, L. Rodríguez, F. Casacuberta, and I. García-Varea. Interactive Pattern Recognition. In *Proc. of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (ICMI-MLMI'97)*, volume 4892, p. 60–71. 2007.

D. Wells. *Recent economic changes and their effect on the production and distribution of wealth and the well-being of society*. D. Appleton and company, 1899.

P. Whitelock, M. Wood, B. Chandler, N. Holden, and H. Horsfall. Strategies for interactive machine translation: the experience and implications of the UMIST Japanese project. In *Proc. of the 11th Coference on Computational Linguistics (COLING'86)*, p. 329–334, 1986.

Chapter 2

Representation, Applications and Corpora

Chapter Outline

In this chapter, we will introduce the corpora that will be used in the remainder of this thesis. First, we will use a unified representation of the search space, the *word graph* (WG). In Section 2.2, the corpora for the interactive techniques will be described. Along with the description of each task, we will give an explanation of how to interpret the WGs for the specific problem. Finally, the evaluation metrics for interaction will be defined. On the other hand, Section 2.3 will show the corpora used as input modalities in the multimodal experiments. The evaluation metrics for multimodal interaction will also be defined.

2.1 Unified representation for SP

In Section 1.3, we introduced Equation (1.7) as the fundamental equation to SP, upon which other approaches to SP are built (e.g., Equation (1.9)). Equation (1.7) relies on the computation of the posterior probability. However, the direct computation of this probability is impractical to model in the most interesting problems. Hence, several strategies have been devised to work at element level rather than at structure level [Brown et al., 1993; Rabiner, 1989; Toselli et al., 2010; Zens et al., 2002]. In these approaches, the search algorithm obtains the structured output by exploring a search space constituted from output elements. Here, for the sake of convenience, we will adopt the WGs as a common representation to such search space. WGs can store efficiently the output search space, and there is a sound theoretical framework and algorithms for them. To our advantage, the WGs can be obtained just as a by-product of the traditional search algorithms for each of the problems we tackle in this thesis.

Usually, search algorithms avoid the computation of constants that can be safely ignored from Equation (1.7) thanks to the arg max. For that reason, at this point we will speak of *scores* instead of *probabilities*. Consequently, as they are obtained by the search algorithms, WGs represent a score that is proportional to the posterior probability distribution over the output hypothesis space, for a given input variable $\boldsymbol{x}$ of an input space $\mathcal{X}^*$. Formally, $G(\boldsymbol{x})$ is a WG represented as a directed, acyclic, weighted graph defined by the tuple $G(\boldsymbol{x}) = (Q, q_I, q_F, t_{\boldsymbol{x}}, \mathcal{Y}, A, F_{\boldsymbol{x}})$ where:

- Q is a set of nodes being q_I the initial node and q_F the final node[1].
- $t_{\boldsymbol{x}}(u, v) : Q \times Q \to \mathcal{X}^*$ where $t_{\boldsymbol{x}}(u, v)$ is the set[2] of elements in $\boldsymbol{x}$ covered from nodes u to v.

[1] We will assume, without loss of generality, that there is a single initial node and a single final node. In addition, for convenience, we will assume that the nodes are ordered following a topological order.

[2] In the case that the input-output relation is monotonous the set $t_{\boldsymbol{x}}(u, v)$ can be defined as the start and end indices of a subsequence of $\boldsymbol{x}$. In this case, the start index would be associated to node u whereas the end node would be associated to node v.

- $\mathcal{Y}$ is the set of possible output labels.
- $A : \mathcal{Y} \times Q \times Q$ is a set of edges, $e = (y, u, v)$, where y is an output label that is generated from a start node u to an end node v[3]. Consequently, y is responsible for explaining the subset of $\boldsymbol{x}$ given by $t_{\boldsymbol{x}}(u, v)$. We say that y is *aligned* to $t_{\boldsymbol{x}}(u, v)$.
- $F_{\boldsymbol{x}}(e) : A \to \mathbb{R}$ is a score function that evaluates how likely is y to be generated by $t_{\boldsymbol{x}}(u, v)$ in the context of nodes u and v.

A path $\boldsymbol{e} = (e_1, \ldots, e_k, \ldots, e_K)$ with $e_k = (y_k, u_k, v_k)$ is a sequence of connected edges that represents a complete output hypothesis. Hence, a path must meet the following condition:

$$u_1 = q_I \quad \wedge \quad \forall k, 1 < k \leq K \colon \; v_{k-1} = u_k \quad \wedge \quad v_K = q_F$$

Also, the input coverage of the edges in a path must not present overlapping and they must cover the whole input $\boldsymbol{x}$. Therefore, a WG must also accomplish the following conditions:

a) $\forall (e_i, e_j) \in \boldsymbol{e}, e_i \neq e_j \colon \; t_{\boldsymbol{x}}(u_i, v_i) \cap t_{\boldsymbol{x}}(u_j, v_j) = \emptyset$
b) $\cup_{e_k \in \boldsymbol{e}} \, t_{\boldsymbol{x}}(u_k, v_k) = \boldsymbol{x}$

Finally, the score of a path can be obtained as the product of the edge scores along the path,

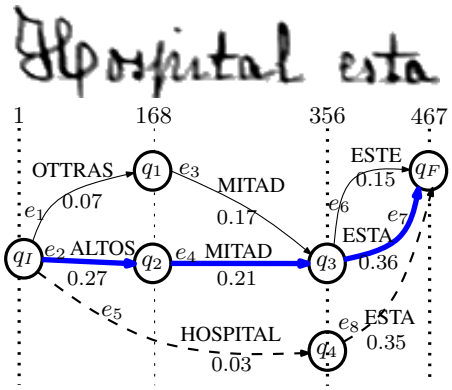
$$F_{\boldsymbol{x}}(\boldsymbol{e}) = \prod_{k=1}^{K} F_{\boldsymbol{x}}(e_k) \tag{2.1}$$

A detail of a WG for a handwritten text recognition problem is shown in Figure 2.1. The figure represents the search space for the handwritten text 'Hospital esta'. In addition, some of the variables that define the WG are instantiated in the left part of the figure. Two paths have been highlighted: in bold, the most likely path; in dashed, the correct path.

With Equation (2.1), we can apply the MAP decision rules for SP and ISP. However, in the way that search algorithms generally work, $F_{\boldsymbol{x}}(\boldsymbol{e})$ is not an actual probability, but a score. In this thesis we find it convenient to convert these scores into probabilities, though this is not strictly necessary to apply the MAP decision rule. Hence, the posterior probability of a path in a WG can be computed as

$$p(\boldsymbol{e} \mid \boldsymbol{x}) = \frac{F_{\boldsymbol{x}}(\boldsymbol{e})}{\sum_{\boldsymbol{e'}} F_{\boldsymbol{x}}(\boldsymbol{e'})} \tag{2.2}$$

[3]We will assume, without loss of generality, that edges only hypothesize one label. A more general definition of an edge would allow more than one label. However, that representation can be easily transformed to use only one label by splitting the edge in multiple edges.



$$
\begin{aligned}
\boldsymbol{x} &= \{x_1, x_2, \ldots, x_{467}\} \\
Q &= \{q_I, q_1, q_2, q_3, q_4, q_F\} \\
t_{\boldsymbol{x}}(q_I, q_4) &= \{x_1, \ldots, x_{356}\} \\
t_{\boldsymbol{x}}(q_4, q_F) &= \{x_{357}, \ldots, x_{467}\} \\
A &= \{e_1, e_2, \ldots, e_8\} \\
e_5 &= (\text{`}HOSPITAL', q_I, q_4) \\
e_8 &= (\text{`}ESTA', q_4, q_F) \\
F_{\boldsymbol{x}}(e_5) &= 0.03 \\
F_{\boldsymbol{x}}(e_8) &= 0.35 \\
F_{\boldsymbol{x}}(\boldsymbol{e}^* = \{e_5, e_8\}) &= 0.0105
\end{aligned}
$$

Figure 2.1: Example of a WG for a handwritten text recognition problem. Above, the digitized and preprocessed text image for the handwritten text 'Hospital esta', that is represented by a feature vector $\boldsymbol{x}$ with 467 input elements. The vertical dotted lines are used to align the start and end of each edge with the corresponding segment of $\boldsymbol{x}$, where the indices of the vector are indicated by the numbers on top of the vertical lines. The WG consists of 8 nodes and 6 edges. Each edge e_k also displays, the output label and the score $F_{\boldsymbol{x}}(e_k)$. The most likely path, $\hat{\boldsymbol{e}} = \{e_2, e_4, e_7\}$ is represented by the bold edges, whereas the correct path, $\boldsymbol{e}^* = \{e_5, e_8\}$, is displayed with dashed edges.

where the denominator accounts for probability mass of all the paths in the WG. Equation (2.2) can be efficiently computed based on the well-known *forward-backward*-like algorithm [Wessel et al., 2001].

Now that we have the probability of a path, we can compute the probability of a sequence of words. Given that WGs can be ambiguous[4], in general, there may be more than one path associated with the same output sequence $\boldsymbol{y}$. Let $w(\boldsymbol{e})$ be the sequence of labels associated with $\boldsymbol{e}$. The probability of the word sequence $\boldsymbol{y}$ is computed as:

$$
p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{\boldsymbol{e}: w(\boldsymbol{e}) = \boldsymbol{y}} p(\boldsymbol{e} \mid \boldsymbol{x}) \tag{2.3}
$$

Then, in the MAP decision rule for the SP problem, the word sequence with maximum probability can be obtained as:

$$
\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \sum_{\boldsymbol{e}: w(\boldsymbol{e}) = \boldsymbol{y}} p(\boldsymbol{e} \mid \boldsymbol{x}) \tag{2.4}
$$

This maximization problem can be solved by determinizing the WG, which is exponential in the worst case (although in many practical cases the real cost is admissible) [Mohri, 2009]. Nevertheless, typically an adequate solution is to

[4]From the same node several nodes can be reached with the same y.

approximate the sum by its dominant addend, which happens to be the path with highest probability:

$$\hat{\boldsymbol{y}} \approx \arg\max_{\boldsymbol{y}} \max_{\boldsymbol{e}:w(\boldsymbol{e})=\boldsymbol{y}} p(\boldsymbol{e} \mid \boldsymbol{x}) \tag{2.5}$$

Equation (2.5) can be solved more efficiently than Equation (2.4) by means of *dynamic programing* search algorithms [Bellman, 2003; Jelinek, 1997]. Note that the normalization in Equation (2.2) is not necessary, since the denominator can be canceled out by the max in Equation (2.4) or Equation (2.5).

In the same way, the most probable suffix using the MAP decision rule for ISP can be obtained as:

$$\hat{\boldsymbol{y}_s} \approx \arg\max_{\boldsymbol{y}_s} \max_{\boldsymbol{e}:w(\boldsymbol{e})=\boldsymbol{y}_p\cdot\boldsymbol{y}_s} p(\boldsymbol{e} \mid \boldsymbol{x}, \boldsymbol{y}_p) \tag{2.6}$$

where the restriction $\boldsymbol{e} : w(\boldsymbol{e}) = \boldsymbol{y}_p \cdot \boldsymbol{y}_s$ means that the path $\boldsymbol{e}$ must be consistent with the given prefix. It should be mentioned that, in practice, it is not always possible to find a path in the WG that meets this condition with probability larger than 0. For this reason, the restriction is typically relaxed to find the path with shorter edit distance with respect to the prefix [Barrachina et al., 2009; Koehn, 2009].

Although ideally a WG can represent the whole set of possible outputs, in practice it is frequently not possible to compute or to store the whole search space. Still, WGs are useful for characterizing a subset of the most likely solutions from the hypothesis space. First, WG can encode hypotheses in a much more compact way than traditional n-bests lists. Second, there exists a reasonable collection of well-defined and well-known efficient algorithms for them [Mohri, 2009; Vidal et al., 2005a,b], as they can be seen as a particular case of weighted *finite-state machines* (FSM). The hypotheses encoded in the WG are those whose probability is large enough, according to the search algorithm used to decode the input signal [Liu and Soong, 2006; Ortmanns et al., 1997; Toselli et al., 2011; Ueffing et al., 2002].

2.2 Structured prediction tasks

2.2.1 Optical character recognition

OCR is the conversion of scanned images of handwritten or printed characters into actual computer characters. Typically these characters can be easily isolated so the problem is transformed in a sequence labeling problem. Many algorithms have been used to solve the problem of isolated character recognition effectively[5]. Similarly to the example in Section 1.5, the OCR problem considered for this thesis consists in a series of digits and a control code: the Spanish

[5]See http://yann.lecun.com/exdb/mnist/ for a list of techniques.

national identification number. The corpus is a compilation of handwritten *national identification numbers* (DNI, from Spanish *documento nacional de identidad*) from real paper forms acquired by the RIVA group in the Institut Tecnològic d'Informàtica[6]. The training is composed by $1.8M$ handwritten characters for training, and a separate set of $10k$ DNIs, $5k$ for validation and $5k$ for test. See Table 2.1 for more detailed statistics.

	Number of samples	Baseline error rate
DNI	5,262	22.2
Digits	42,096	2.10
Letters	5,262	10.05
Total chars	47,358	2.86

No. errors in DNI	0	1	2	3	4	5	6	7	8
No. DNIs	4,094	972	162	28	3	0	2	1	0

Table 2.1: Some statistics regarding the OCR DNI corpus.

Each handwritten DNI number, $\boldsymbol{x} = \{x_1, \ldots, x_9\}$, is a series of 9 images of handwritten characters. The images correspond, one-to-one, to the characters of a DNI, $\boldsymbol{y} = \{d_1, \ldots, d_8, c\}$, which consists of 8 digits $d_1 \ldots d_8$ plus a letter c from a list of control characters C, $c \in \boldsymbol{C}$. The control character can be computed from the digits using Algorithm 1.

Algorithm 1: Algorithm to obtain the error code for a given number for Spanish DNI numbers.

Input: $d_1 \ldots d_8$
Output: c
$\boldsymbol{C} \leftarrow$ "TRWAGMYFPDXBNJZSQVHLCKE";
return $\boldsymbol{C}[d_1 \ldots d_8 \mod 23]$

As we explained in Equation (1.9), the noisy channel approach allows us to model explicitly the output structure and the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$. In the DNI problem, $\Pr(\boldsymbol{y})$ can be modeled as *finite state machine* (FSM) that computes the modulo of a number. In that FSM, for a given base B and modulo M, we can reach state v from a state u with y if $v = (u * B + y) \mod M$ (see Figure 2.2 for a FSM with $B = 10$, $M = 2$ and $C =$ "AB"). The language model probability can be then obtained by the product of the transition probabilities the model passes through. The transition probability from state u to state v, $p(v \mid u)$, follows a uniform probability distribution. In the DNI case, as we use digits $B = 10$ and $M = 23$ since we have 23 control characters. As each of the character image representation in $\boldsymbol{x}$ can be considered as independent from each other, $\Pr(\boldsymbol{x}|\boldsymbol{y})$ can be modeled with a

[6] https://prhlt.iti.upv.es/w/isl

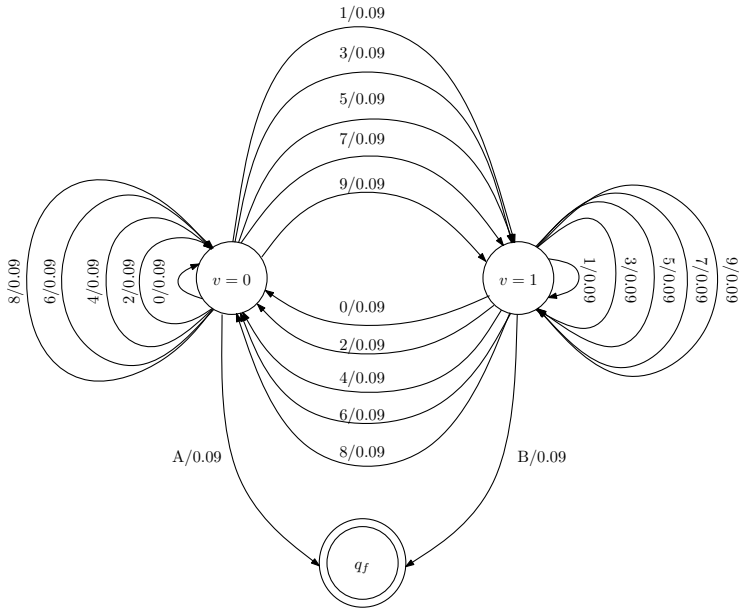


Figure 2.2: An example of a DNI language model with $B = 10$, $M = 2$ and $\boldsymbol{C}$ = "AB". In each state, v indicates the modulo for the processed digits, where $v = 0$ means the number is even and $v = 1$ means that it is odd. Thus, in this simple example, even numbers go to state $v = 0$ and odd numbers to the state $v = 1$. Then, when the control code arrives it only accepts 'A' if the number is even and 'B' if it is odd.

naive Bayes assumption as $\prod_{i=1}^{|\boldsymbol{x}|} \Pr(x_i|y_i)$. $\Pr(x_i|y_i)$ can be, after some simple probability transformations, approximated by the posterior probability, $p(y_k \mid x_k)$, of a k *nearest neighbor* (kNN) classifier [Goldberger et al., 2004]. Since the language model probability is uniform, the posterior probability and the likelihood are proportional. In summary, the score for an edge in this problem can be defined as

$$F_{\boldsymbol{x}}\left((y_k, u_k, v_k)\right) = p(y_k \mid t_{\boldsymbol{x}}(u_k, v_k)) \tag{2.7}$$

where the FSM transition probabilities can be ignored since they are constant, and $t_{\boldsymbol{x}}(u_k, v_k) = x_k$.

An example of WG for a DNI with $B = 10$, $M = 2$ and $\boldsymbol{C}$ = "AB" is shown in Figure 2.3. Each state represents the state of the search up to this point. The boxes are highlighted if the correspondent character has been processed. Furthermore, each state has the current state of the module for the processed digits, where $v = 0$ means the the number is even and $v = 1$ means that it is odd.

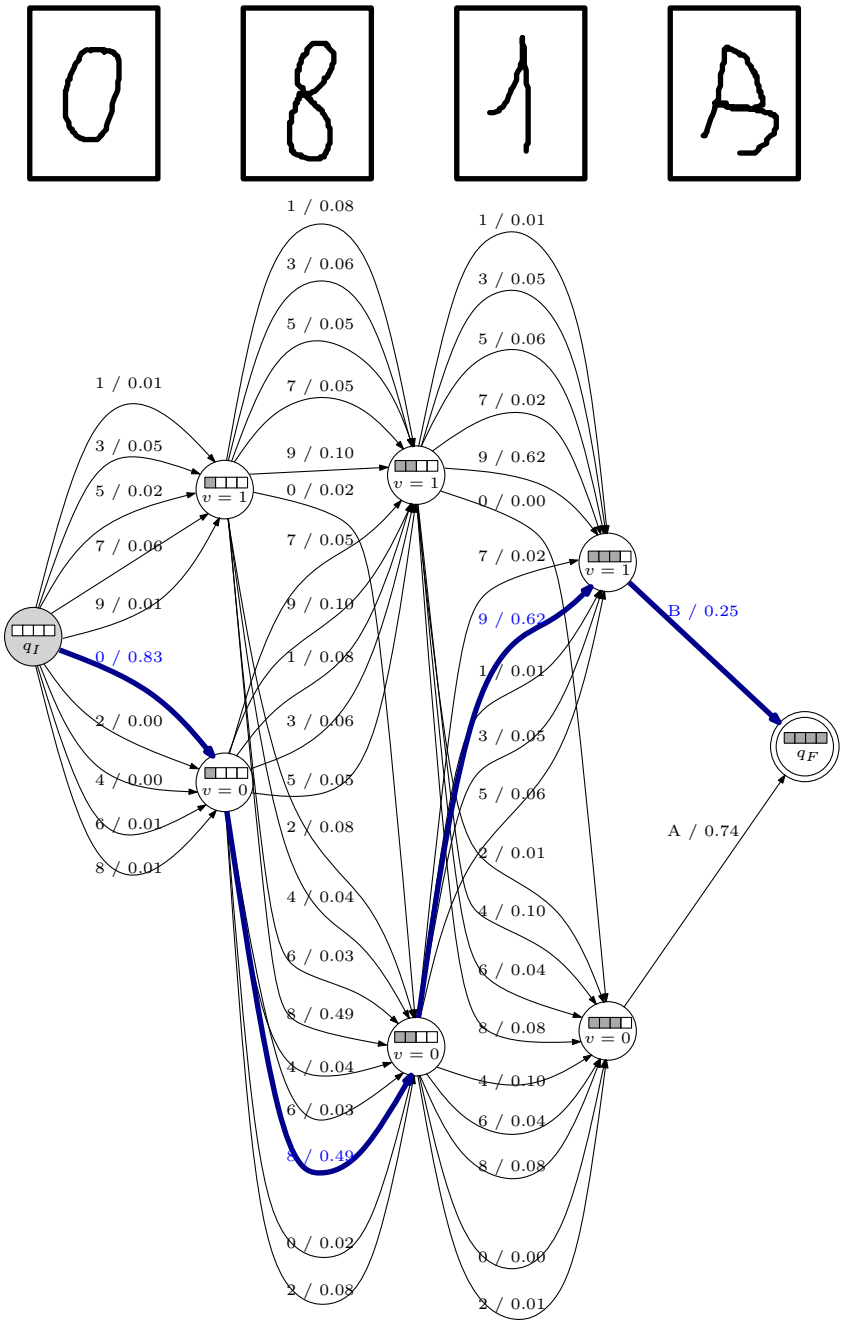
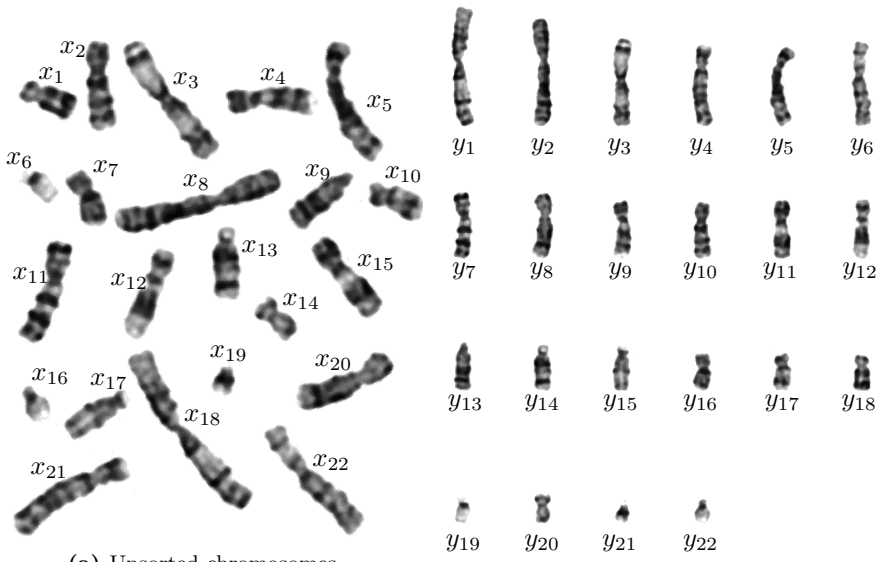


Figure 2.3: An example of WG for a DNI with $B = 10$, $M = 2$ and $\boldsymbol{C}$ = "AB". Each state represents the state of the search up to this point. The gray boxes imply that the correspondent character has been processed. v indicates the current state of the module for the processed digits, where $v = 0$ means the the number is even and $v = 1$ means that it is odd.

2.2.2 The assignment problem

The assignment problem is one of the fundamental combinatorial problems in optimization. It consists in assigning the elements from $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$, one-to-one, to the elements in $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$. In particular, we will consider a simplification of the problem of recognizing human karyotypes [Martínez et al., 2007; Ritter et al., 1995]. A karyotype is the number and appearance of chromosomes in the nucleus of a eukaryote cell. Normal human karyotypes contain 22 pairs of autosomal chromosomes and one pair of sex chromosomes. Normal karyotypes for females contain two X chromosomes, whereas males have both an X and a Y chromosomes. Any variation from the standard karyotype may lead to developmental abnormalities. The chromosomes are depicted (by rearranging a microphotograph, see Figure 2.4) in a standard format known as a karyogram or idiogram: in pairs, ordered by size and position of centromere for chromosomes of the same size. Each chromosome is assigned a label from $\{\text{`}1\text{'}, \ldots, \text{`}22\text{'}, \text{`}X\text{'}, \text{`}Y\text{'}\}$, according with its position in the karyogram.



(a) Unsorted chromosomes.

(b) Chromosomes assigned to their respective classes.

Figure 2.4: To the left the unsorted chromosomes. To the right the chromosomes already classified

Here, we use the "Copenhagen database" [Lundsteen et al., 1980]. For the sake of simplicity, we ignored the initial image segmentation task and assumed that each of the 46 chromosomes in a normal unsorted karyotype is already represented as an individual image. Note that, in this simplification, individual, rather than paired chromosomes are considered and sex chromosomes 'X' and 'Y' are ignored. Moreover, we do not take into account some advances in

karyotype analysis, such as fluorescent dye based spectral karyotyping [Schröck et al., 1996], which allow obtaining colored chromosome images and may significantly simplify the real human karyotyping problem. Then, the problem consists in, given the unsorted karyotype (see Figure 2.4a), obtain the sorted karyotype by assigning the images to the type of chromosome (see Figure 2.4b). The corpus consists of two data sets of 100 karyotypes each (2,200 images in total per each data set) for which the likelihoods $p(t_{\boldsymbol{x}}(u,v)|y)$, obtained from *hidden Markov models*[7] [Martínez et al., 2003], are already given[8]. Although the optimal assignment can be obtained in $O(|\boldsymbol{x}|^3)$ by the Hungarian algorithm [Edmonds and Karp, 1972], here we are interested in obtaining a WG representing a set of hypothesis. Therefore, we solved the problem with a dynamic programming algorithm with hypothesis pruning. As in the DNI problem, the language model follows a uniform probability distribution. In this case, however, the nodes of the WG represent which input and output elements have already been assigned. The score for an edge can be defined as in Equation (2.7):

$$F_{\boldsymbol{x}}\left((y,u,v)\right) = p(t_{\boldsymbol{x}}(u,v)|y) \tag{2.8}$$

where $t_{\boldsymbol{x}}(u,v)$ is the element in $\boldsymbol{x}$ assigned to y.

An example of WGs for this tasks is shown in Figure 2.5, where only three images and classes are considered to allow a clearer display. Each label represents the probability of assigning the chromosome to the label. In addition, each node represents the state of coverage of images and chromosome classes: two rows of bit vectors are shown where the boxes in gray indicate that the image (top) or chromosome class (bottom) has already been assigned.

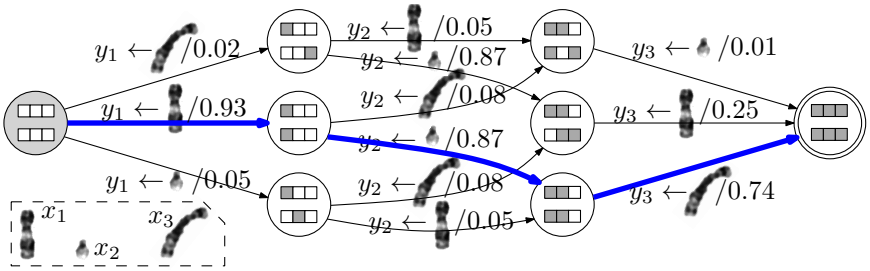


Figure 2.5: Example of WGs for the karyotype problem, where only three images and classes are considered to allow a clearer display. Each label represents the probability of assigning the chromosome to the label. In addition, in each node represents the state of coverage of images and chromosome classes: two rows of bit vectors are shown where the boxes in gray indicate that the image (top) or chromosome class (bottom) has already been assigned.

[7] See Section 2.2.3 for further information regarding *hidden Markov models*.
[8] https://prhlt.iti.upv.es/w/karyo

2.2.3 Automatic speech and handwritten text recognition

Both *automatic speech recognition* (ASR) [Rabiner, 1989] and *handwritten text recognition* (HTR) [Toselli et al., 2004] are typically modeled in the same way. According to Equation (1.9), they can be formulated as the problem of finding the most likely word sequence, $\boldsymbol{y} = (y_1, y_2, \ldots, y_{|\boldsymbol{y}|})$, for a feature vector sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_{|\boldsymbol{x}|})$ describing a text image or speech signal along its corresponding horizontal or time axis. $\Pr(\boldsymbol{y})$ is approximated by a word language model, usually back-off n-grams [Jelinek, 1997], whereas $\Pr(\boldsymbol{x}|\boldsymbol{y})$ is approximated by concatenated HMM phoneme/character models.

n-gram language models

The assumption beyond an n-gram language model is that the conditional probability $\Pr(y_k \mid y_1^{k-1})$ (from a Bayes decomposition of $\Pr(\boldsymbol{y})$) can be modeled using only the history of the last $n-1$ words. However, as the history becomes bigger, their probabilities become also sparser. Thus, it is often the case that a sequence of n words has not been observed in the training phase, and then its probability is zero. Therefore, it is necessary to apply some sort of smoothing technique, where backing-off to lower order models is most successful [Katz, 1987]. As a result, the back-off probability for the language model can be modeled as

$$p_{bo}(y_k \mid y_{k-n+1}^{k-1}) = \begin{cases} \varepsilon(y_{k-n+1}^{k})\, p(y_k \mid y_{k-n+1}^{k-1}) & \text{if } C(y_{k-n+1}^{k}) > t \\ \vartheta(y_{k-n+1}^{k-1})\, p_{bo}(y_k \mid y_{k-n+2}^{k-1}) & \text{otherwise} \end{cases} \tag{2.9}$$

where $p(\cdot)$ is the frequentist probability and $\varepsilon(\cdot)$ is a discount factor for the probability of the n-gram in the case that the counts $C(y_{k-n+1}^{k})$ surpass a certain threshold t. On the other hand, $\vartheta(\cdot)$ is a back-off weight for the n-grams that do not surpass such threshold. All these quantities can be estimated using various techniques, among which interpolated *Kneser-Ney* [Kneser and Ney, 1995] is one of the most popular and top performing approaches [Chen and Goodman, 1996].

Hidden Markov models

Hidden Markov modes (HMMs) are consolidated statistical models for observations with temporal dependences. The systems being modeled by HMMs are assumed to be a Markov process[9] with unobserved (hidden) states[10]. The relationship between the input and the output is monotonic but, at least in the

[9] More specifically first-order Markov process, i.e., the next state depends only on the current state and not on the sequence of events that preceded it.

[10] The sequence of states of the model the observation passes through is unknown. However, the parameters of the model are known.

problems we will deal with, the input is typically much longer than the output. Moreover, the segmentation (the segment of input elements that corresponds to an output) is unknown. Let $\boldsymbol{q} \in \mathcal{Q}$ be a sequence of HMM states linked to $\boldsymbol{x}$, such as each input vector has an associated state, $\boldsymbol{q} = \{q_1, \ldots, q_{|\boldsymbol{x}|}\}$. Let $\boldsymbol{\tau} \in \mathcal{T}, \boldsymbol{\tau} = \{\tau_1, \ldots, \tau_{|\boldsymbol{y}|}\}$ a sequence of states that represent the alignment between the words in $\boldsymbol{y}$ and the vectors in $\boldsymbol{x}$, so that y_1 is aligned to $x_0^{\tau_1}$, y_2 is aligned to $x_{\tau_1+1}^{\tau_2}$, etc. We can sum up over all possible sequences of states and segments,

$$\Pr(\boldsymbol{x}|\boldsymbol{y}) = \sum_{\boldsymbol{\tau} \in \mathcal{T}} \sum_{\boldsymbol{q} \in \mathcal{Q}} \Pr(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{\tau} \mid \boldsymbol{y}) \tag{2.10}$$

Now, we expand $\Pr(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{\tau} \mid \boldsymbol{y})$ using the chain rule and apply the Markov assumptions. Then, we can approximate Equation (2.10) by

$$\Pr(\boldsymbol{x}|\boldsymbol{y}) \approx \sum_{\boldsymbol{\tau} \in \mathcal{T}} \sum_{\boldsymbol{q} \in \mathcal{Q}} \prod_{n=1}^{|\boldsymbol{y}|} p(x_{\tau_{n-1}+1}^{\tau_n}, q_{\tau_{n-1}+1}^{\tau_n} \mid y_n) \tag{2.11}$$

where $p(x_{\tau_{n-1}+1}^{\tau_n}, q_{\tau_{n-1}+1}^{\tau_n} \mid y_n)$ can be decomposed by assuming again that $p(x_m \mid q_1^m, y_n)$ is Markovian and does not depend on y_n

$$p(x_{\tau_{n-1}+1}^{\tau_n}, q_{\tau_{n-1}+1}^{\tau_n} \mid y_n) \approx \prod_{m=\tau_{n-1}+1}^{\tau_n} p(q_m \mid q_{m-1}, y_n) \quad p(x_m \mid q_m) \tag{2.12}$$

Now, $p(q_m \mid q_{m-1}, y_n)$ is the transition probability from state q_{m-1} to state q_m from the lexical model of word y_n. Each lexical word is modeled by a probabilistic FSM, which represents all possible concatenations of individual phonemes/character to compose the actual word. The lexical HMM is obtained by composition of the phoneme/character HMMs into the edges of this automaton. Next, in each phoneme/character a Gaussian mixture per state is used to model $p(x_m \mid q_m)$. This mixture serves as a probabilistic law to the emission of feature vectors on each model state. The optimum number of HMM states and Gaussian densities per state are tuned empirically.

The model parameters can be easily trained from samples (handwritten text image or speech utterance) accompanied by the transcription of these samples into the corresponding sequence of phonemes/characters. This training process is carried out by using a well known instance of the EM algorithm called forward-backward or Baum-Welch [Baum et al., 1970]. In this thesis we have used the HTK software [Young et al., 2006] to train HMMs. The principal difference between ASR and HTR lays in the type of feature vectors: while in the case of ASR they are acoustic data, the input sequences for off-line HTR represent line-image features. Figure 2.7 shows an example of how a HMM models two feature vector subsequences pertaining to the phoneme "a" and the character "a".

Once all the phoneme/character, word and language models are available, recognition of new test sentences can be performed. Thanks to the homogeneous finite-state nature of all these models, they can be easily integrated into a single global model on which a search process is performed for decoding the input feature vectors sequence. An example of WG for HTR, which has a more visual representation, is shown in Figure 2.6. In this case, a word 2-gram was used as a language model. Thus, each state is represented by the word preceding it, and the index of the input vector where the next word to be decoded starts. Scores have been omitted for simplicity.

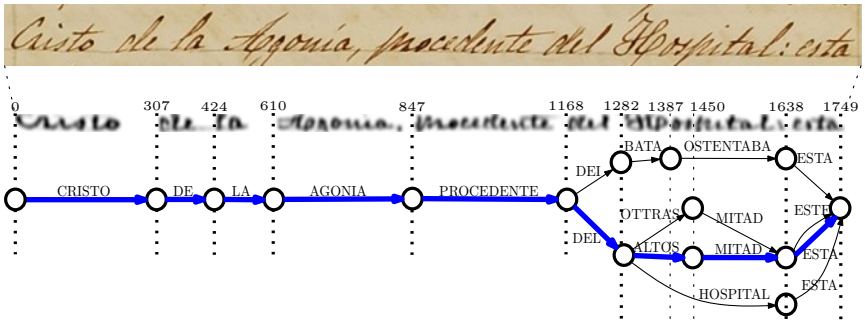


Figure 2.6: Word graph for the handwritten sentence 'Cristo de la Agonía procedente del Hospital está'. Each state is represented by the word preceding it (2-gram), and the index of the input vector where the next word to be decoded starts. Scores have been omitted for simplicity.

In HMMs, the score is defined at word level by using the probability in Equation (2.11) as a logarithm,

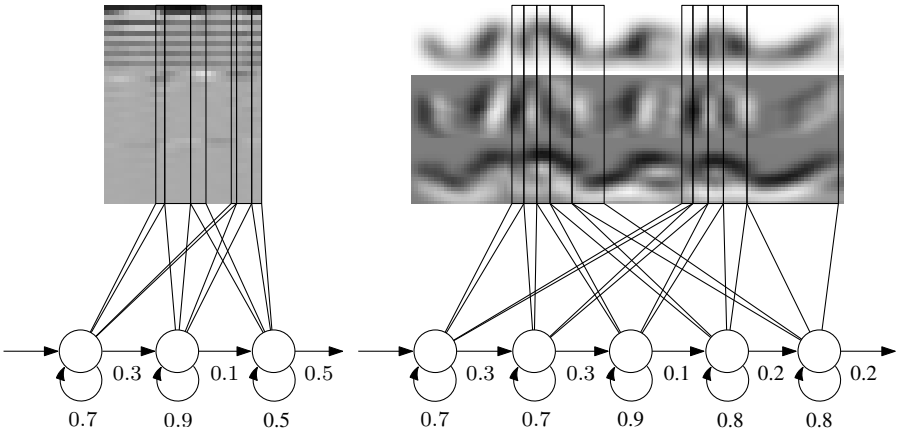
$$F_{\boldsymbol{x}}\left((y_k, u_k, v_k)\right) = \exp\Big[\alpha \ \log p\left(t_{\boldsymbol{x}}(u_k, v_k) \mid y_k\right) + \beta \ \log p_{bo}(y_k \mid y_{k-n+1}^{k-1}) + \rho\Big] \tag{2.13}$$

where $t_{\boldsymbol{x}}(u, v) = (x_{\tau_{n-1}+1}^{\tau_n}, q_{\tau_{n-1}+1}^{\tau_n})$. The parameter α, which is usually set to 1, is used to scale the likelihood. On the other hand, β scales the probability of the language model, so as to help to compensate the differences in accuracy and description power of both probabilities. In addition, β is also used to cope with the difference in range of the quantities, since the language model probability needs to be scaled to match the dynamic range of the likelihood. Finally, the last term,[11] ρ, is the word insertion penalty or word deletion penalty, depending on the sign. It is used to control the length of the final output. These parameters (α, β, and ρ) are empirically tuned to optimize accuracy on a development set.

[11] Named where it is used as word insertion penalty or word deletion penalty, depending on the sign.

Automatic Speech Recognition

In ASR, each input vector $\boldsymbol{x}$ represents a speech signal, typically a spoken sentence. First, the speech signal is digitized by means of a analog-to-digital converter from a computer microphone. Next, the digitalized signal is transformed into $\boldsymbol{x}$ by extracting a series of features from it. The feature extraction of the ASR system is based on the Mel cepstral coefficients [Rabiner, 1989]. Speech preprocessing reproduces the standard steps for speech recognition. The audio signal is captured from a microphone at 16kHz and digitalized. A sliding window with overlapping is passed over the signal. For each window the following procedure is carried out. First, in the pre-emphasis step, a high-pass filter is used to compensate the differences between high and low frequencies. Second, a Hamming window is applied to smooth out the borders of the window. Next, the signal is converted from the time domain to the frequency domain by means of a discrete Fourier transform. To mimic the mechanism of the human ear, the Mel scale is used to group the energy of frequencies that are indistinguishable to humans. Then, volume normalization is carried out by applying a logarithmic transformation. Now, a discrete cosine transform is performed, resulting in the so-called cepstral coefficients. The frame energy is added as an extra element. This value is a global measure for the frame and it is computed as the first element of the discrete cosine transform. Finally, first and second derivatives are added to the final feature vector. Figure 2.7a shows an example of ASR feature vectors for the word "saca", and how they can be aligned with the HMM states for the phoneme "a".



(a) HMM-ASR **(b)** HMM-HTR

Figure 2.7: Example of 3-states HMM for ASR (left) and 5-states HMM for HTR (right) modeling (sequences of feature vectors) instances of the phoneme "a" and the character "a", respectively, within the Spanish word "saca". The states are shared among all instances of phonemes/characters of the same class.

The experiments were performed using the Wall Street Journal (WSJ) corpus [Pallett et al., 1994]. The ARPA WSJ corpus consists of samples of read texts drawn from WSJ publications recorded under high-quality conditions. Up to 81 hours of training material (WSJ0+WSJ1 partitions) were used to train speaker independent HMMs with HTK. HMMs were word-internal triphones and gender independent. They were composed of three emitting states (24 gaussians per state) and a left-to-right topology with self loops. Silence and inter-word silence models were trained. The test was composed of 213 sentences and $3.4k$ running words with a perplexity of 168. The recognition was performed with the open vocabulary setup ($64k$ words). Still, the test set contained 314 OOVs. A summary of this corpus can be found in Table 2.2.

	Training
Sentences	37,394
Speakers	284
Triphones	11,889
Tied-states	5,602
Densities	134,502

(a) Training.

	Dev	Test
Sentences	403	213
Running words	6,721	3,446
Speakers	10	10
OOV (%)	3.9	1.7
Perplexity	150	149

(b) Test and development.

Table 2.2: Summary of statistics of the WSJ corpus.

Handwritten Text Recognition

The HTR problem is formulated and modeled in a very similar fashion to the ASR problem. In this case, however, $\boldsymbol{x}$ represents a line of digitized manuscript. The HTR system used here follows the classical architecture composed of three main modules: document image preprocessing, line image feature extraction and HMM training/decoding [Toselli et al., 2004]. The following steps take place in the preprocessing module. First, the skew of each page is corrected; we understand "skew" as the angle between the horizontal direction and the direction of the lines on which the writer aligned the words. Then, a conventional noise reduction method is applied on the whole document image, whose output is then fed to the text line extraction process which divides it into separate text lines images. Finally, slant correction and size normalization are applied on each separated line. A more detailed description of the feature extraction can be found in [Toselli et al., 2004] and [Romero et al., 2007].

As our HTR system is based on HMMs, each preprocessed line image is represented as a sequence of feature vectors. To do this, the feature extraction module applies a grid to divide the text line image into $N \times M$ squared cells. From each cell, three features are calculated: normalized gray level, horizontal gray level derivative and vertical gray level derivative. The way these three features are determined is described in [Toselli et al., 2004]. Columns of cells or *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 $3N$-dimensional feature vectors is obtained. In Figure 2.7b an example of the sequence of feature vectors for the word "saca" is shown graphically.

HTR experiments were conducted on the "Cristo-Salvador" corpus [Romero et al., 2007]. This corpus was compiled from the legacy handwriting document from the XIX century, which was kindly provided by the *Biblioteca Valenciana Digital* (BIVALDI)[12]. This is a rather small document composed of 53 color images of text pages. Some of these page images are shown in Figure 2.8. In the *page* version of this corpus, the test set is formed by 491 samples corresponding to the last ten lines of each document page ($4.5k$ running words), whereas the training set is composed of the 681 remaining samples ($6.4k$ running words). The experiments were run on a closed vocabulary containing $3.4k$ words. Note that the small training ratio, 2.8 training words per lexicon entry in average, results in a high test perplexity of 360 for a 2-gram language model. All the information related with page partitions is summarized in Table 2.3.

Figure 2.8: Examples of corpus "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	16	2.8
Characters	36,729	25,487	62,216	78	0	470

Table 2.3: Basic statistics of the partition *page* of the database Cristo-Salvador

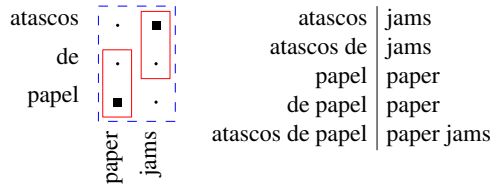
[12] http://bv2.gva.es

2.2.4 Machine Translation

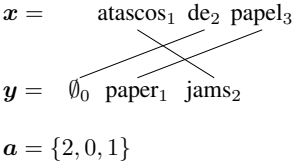
Machine translation (MT) essentially consists on, given a sentence in a source language $\boldsymbol{x}$, to obtain a sentence in a target language $\boldsymbol{y}$ that is a translation of $\boldsymbol{x}$. When applying Equation (1.9) to statistical MT, $\Pr(\boldsymbol{y})$ is modeled as an n-gram language model, whereas $\Pr(\boldsymbol{x}|\boldsymbol{y})$ can be approximated by word-based models [Brown et al., 1993]. On the other hand, log-linear phrase-based models [Koehn et al., 2003; Tomás and Casacuberta, 2001; Zens et al., 2002], which are built upon word-based models, follow Equation (1.7). The former are good for obtaining the alignments between source and target words, but they are rather limited concerning translation quality since they cannot model contextual information properly. In contrast, phrase-based models may achieve good levels of quality for many translation tasks.

Brown et al. [1993] approached the problem of word-based MT from a statistical point of view, by introducing a hidden variable, $\boldsymbol{a} \subseteq \{1, \dots, |\boldsymbol{x}|\} \times \{1, \dots, |\boldsymbol{y}|\}$, where $a_{j,i} = 1$ indicates that the source word x_j is aligned to the target word y_i, and $a_{j,i} = 0$ indicates the contrary (see Figure 2.9a for a visual illustration). This alignment matrix allows all possible alignment patterns. Nevertheless, the huge number of possibilities, $2^{|\boldsymbol{x}||\boldsymbol{y}|}$, makes this approach impractical. Thus, Brown et al. [1993] decided to constrain the alignments for their word-based models to $\boldsymbol{a} : \{1, \dots, |\boldsymbol{x}|\} \to \{0, \dots, |\boldsymbol{y}|\}$. Here, $a_j = i$ represents the source word x_j is aligned to the target word y_i, and $a_j = 0$ means that x_j is not aligned to any target word (a visual representation in Figure 2.9b). Formally, we can marginalize over the set of all possible alignments between the words in $\boldsymbol{x}$ and the words in $\boldsymbol{y}$,

$$\Pr(\boldsymbol{x} \mid \boldsymbol{y}) = \sum_{\boldsymbol{a}} \Pr(\boldsymbol{x}, \boldsymbol{a} \mid \boldsymbol{y}) \tag{2.14}$$



(a) Display of the alignment matrix and extraction of consistent segments. A ■ indicates that the source and target words are aligned. Normal and dashed boxes constitute phrases extracted from coherent alignments, which are explicitly represented to the right.



(b) Representation of the alignment variable in word-based models. Subscripts indicate word position whereas $\emptyset_0$ is the null alignments, i.e., the source word is not aligned to any output word.

Figure 2.9: Different visualization of alignments in MT.

Then, $\Pr(\boldsymbol{x}, \boldsymbol{a}|\boldsymbol{y})$ can be decomposed using the chain rule,

$$\Pr(\boldsymbol{x}, \boldsymbol{a} \mid \boldsymbol{y}) = \Pr(|\boldsymbol{x}| \mid \boldsymbol{y}) \prod_{j=1}^{|\boldsymbol{x}|} \Big[\Pr(a_j \mid a_1^{j-1}, x_1^{j-1}, |\boldsymbol{x}|, \boldsymbol{y}) \Pr(x_j \mid a_1^{j}, x_1^{j-1}, |\boldsymbol{x}|, \boldsymbol{y}) \Big] \tag{2.15}$$

where the first term models the length of $\boldsymbol{x}$, the second one models the alignment probability, and the third one the translation probability.

The different word *alignment-based* models 1, 2 and HMM are built upon Equation (2.15) by making different assumptions about the distribution probabilities, specially regarding alignment probability. The goal is to make model estimation and search tractable. *Fertility-based* word models 3, 4 and 5 introduce additional concepts like fertility and distortion but their introduction is out of the scope of this section. In word alignment-based models, first, $\Pr(|\boldsymbol{x}| \mid \boldsymbol{y})$ is approximated by $p(|\boldsymbol{x}| \mid |\boldsymbol{y}|)$. Then, $\Pr(x_j \mid a_1^j, x_1^{j-1}, |\boldsymbol{x}|, \boldsymbol{y})$ is approximated by a word-by-word statistical translation dictionary $p(x_j|y_{a_j})$. Nonetheless, it is in the alignment probability where the alignment models differentiate. For instance, in Model 1 the alignment probability is modeled by a uniform probability, $(|\boldsymbol{s}|+1)^{-1}$. Model 2 goes a step further and assumes that the alignment probability is conditioned on the length of source and target sentences, and the source position, $p\left(a_j|j, |\boldsymbol{x}|, |\boldsymbol{y}|\right)$. Finally, in HMM models, the alignment probability is assumed to be a Markov process so the alignment depends on the previous alignment but not on the source position, $p\left(a_j|a_{j-1}, |\boldsymbol{x}|, |\boldsymbol{y}|\right)$.

With respect to parameter estimation, the translation probability is essentially the relative frequency of word x_j being aligned with y_{a_j}. On the other hand, the alignment probability, in Model 2 for instance, can be approximated by the relative frequency of position j in the source sentence to be aligned with position a_j in the target sentence for the given sentence lengths. Nonetheless, these frequencies cannot be estimated directly since the real alignments are unknown. Thus, the EM algorithm is needed to reliably estimate these probabilities [Brown et al., 1993].

As stated before, word-based MT does not achieve good translation quality. As they assume that a source word can only generate one target word, the context where the translation takes place cannot be modeled properly. Phrase-based models aim at solving this issue by working at *phrase* level instead of *word* level. In phrase-based models the basic units are phrases instead of words. However, we do not know neither the number of phrases nor how they are segmented, as it is not a deterministic problem. Thus, first we need to marginalize over all possible number of phrases $K : K \leq \min(|\boldsymbol{x}|, |\boldsymbol{y}|)$. Then, we uncover the segmentation of $\boldsymbol{y}$ in K phrases as μ_1^K, and the segmentation of $\boldsymbol{x}$ in K phrases as γ_1^K. Finally, in a similar way to Equation (2.14), the alignments between the phrases are explained by α_1^K, which are defined as $\boldsymbol{a}$ but at phrase level. Then,

the likelihood in Equation (1.9) can be approximated in a way that resembles very much that of HMM word-based models:

$$\Pr(\boldsymbol{x} \mid \boldsymbol{y}) \approx \sum_{K} \sum_{\mu_1^K} \sum_{\gamma_1^K} \left[p(|\boldsymbol{x}| \mid |\boldsymbol{y}|) \sum_{\alpha_1^K} \prod_{k=1}^{K} p(\alpha_k \mid \alpha_{k-1}) p(\boldsymbol{x}_{\gamma_{\alpha_{k-1}}}^{\gamma_{\alpha_k}} \mid \boldsymbol{y}_{\mu_{k-1}}^{\mu_k}) \right] \tag{2.16}$$

Note that the bracketed part of Equation (2.16) is equivalent to the HMM instantiation of Equation (2.14) if we replace words by phrases. Nevertheless, as in this case, the number of phrases and the source and target segmentations are unknown, we first need to split $\boldsymbol{x}$ and $\boldsymbol{y}$ into phrases.

The estimation of the probabilities in Equation (2.16) can be carried out by the EM algorithm [Andrés-Ferrer and Juan, 2009]. However, in practice an heuristic approach can be used with good results:

1. Word-based MT systems are estimated in both directions.
2. The alignments from both directions are combined to form an alignment matrix.
3. All coherent phrases are extracted from the alignment matrix up to a certain phrase length.
4. The translation probabilities are obtained by the relative frequencies.

Figure 2.9a is an example of how coherent phrases are extracted from the alignment matrix. Regarding the alignment probability, phrases already support intra-phrase reordering. In consequence, for language pairs that usually do not present long reorderings, a monotonous approach can be followed. In this case, the alignments are known so the alignment variable can be canceled out from Equation (2.16).

Finally, the translation probability can be combined with the inverse translation probability, the language model, and other phrase features in a log-linear manner to model the posterior probability in Equation (1.7):

$$\Pr(\boldsymbol{x} \mid \boldsymbol{y}) \approx \frac{\exp\left(\sum_f \lambda_f h_f(\boldsymbol{x}, \boldsymbol{y})\right)}{Z(\boldsymbol{x})} \tag{2.17}$$

where h_f are feature functions (the translation log-probabilities and the log-probabilities of the language model among others), λ_f are the scaling factors for each feature function, and $Z(\boldsymbol{x})$ is a normalization factor. The scaling factors are typically obtained with a development set to optimize the loss function of interest. The normalization factor is a constant that can be canceled out in the MAP decision rule. Therefore, the score of an edge in log-linear phrase-based

models can be expressed as

$$F_{\boldsymbol{x}}\left((y_k, u_k, v_k)\right) = \exp\left[\sum_f \lambda_f h_f\left(t_{\boldsymbol{x}}(u_k, v_k), y_k\right)\right] \tag{2.18}$$

For more details on learning and searching in phrase based models see [Koehn, 2010].

The experiments on MT were conducted on the Xerox corpus [Esteban et al., 2004], since it has been extensively used in the literature to obtain IMT results. The Xerox corpus is a collection of technical manuals in English, Spanish, French, and German. The English version is the original document, while the others are professional translations of the original. The English and Spanish documents were used in this work. The corpus features approximately $0.7M$ running words for training and $90k$ running words for test. Language model perplexities for test are 48 for English and 33 for Spanish. Examples of sentence pairs are shown in Figure 2.10 and some statistics are summarized in Table 2.4. In addition, publicly available log-linear decoders, such as *Moses* [Koehn et al., 2007] and *Thot* [Ortiz et al., 2005], are able to produce word graphs (see Figure 2.11 for an example).

English	Spanish
if a particular capability is not available in your network environment , the option will not appear in the dialog .	si alguna de las funciones no se encuentra disponible en su entorno de red , no aparecerá en el cuadro de diálogo .
use this button to expand the search for xerox devices .	use este botón para ampliar la búsqueda de dispositivos xerox .
the search may be expanded to include additional snmp community names that have been added to your network .	la búsqueda puede ampliarse para incluir otros nombres de comunidades de snmp que se han agregado a la red .
show devices enables you to limit the number of document centres displayed in the list .	mostrar dispositivos permite limitar el número de sistemas document centre que se muestran en la lista .

Figure 2.10: Examples of sentence pairs for the Xerox corpus

		English	Spanish
Training	Sentences	55,761	55,761
	Running words	665,400	752,607
	Vocabulary	7,957	11,051
	Perplexity	14.37	13.63
Test	Sentences	1,125	1,125
	Running words	8,370	10,106
	OOVs (%)	3.9	5.9
	Perplexity	48.28	32.92

Table 2.4: Statistics for the Xerox corpus

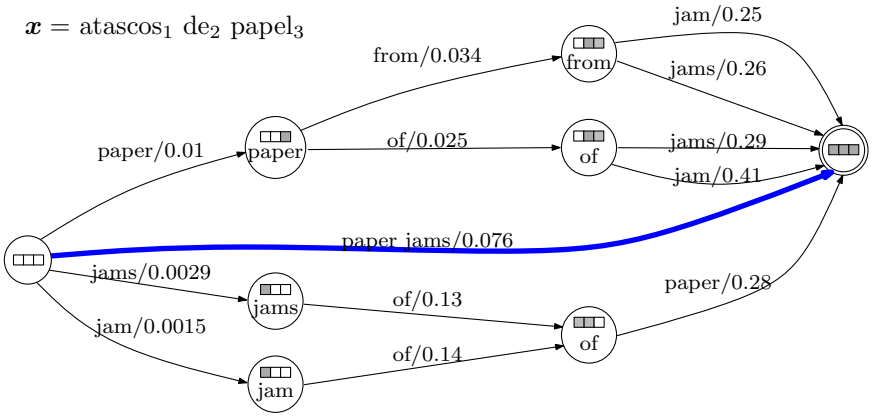


Figure 2.11: Example of translation WG for the Spanish source sentence "atascos de papel". Each state is defined by the previous 2-gram history and the source coverage vector, which identifies what input words have been translated up to the given state.

2.3 Corpora for multimodal interaction

2.3.1 On-line handwritten text for IMT

On-line HTR consists in recognizing handwritten text from pen strokes (or ink) in lieu of scanned images. The on-line HTR problem can be formulated just like the off-line HTR problem. Nevertheless, in on-line HTR the feature vectors are obtained differently. In this work, we follow the feature extraction approach from Pastor et al. [2005]. First, the strokes are preprocessed by eliminating pen-up point and consecutively repeated points. Then, a low pass filter is applied to reduce noise. From the resulting trajectory, 6 features are extracted:

1. The vertical position is normalized by scaling and translating it to $[0, 100]$ keeping aspect ratio.

2. The first and second derivatives for the vertical and horizontal position.

3. The curvature, which is the inverse of the radius of the curve in each point.

Regarding the on-line HTR data, the UNIPEN corpus [Guyon et al., 1994] was used. The training data was composed of symbols, digits and the 1,000 most frequent English and Spanish words. The words were generated by concatenating different instances of characters from the same writer, with a total of 17 different writers. Overall, 68 character classes and a total of $23.5k$ unique character instances were used to generate all the $43.8k$ training samples. These samples were used to train the morphological models, which were represented by continuous density left-to-right character HMMs with Gaussian mixtures and variable number of states per character.

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, lower case letters and symbols. The simulation of user interaction was performed in the following way. First, we ran an off-line simulation for keyboard-based IMT. As a result, a list of words which the system failed to predict was obtained. Supposedly, this would be the list of words that the user would correct with handwriting. Then, from UNIPEN corpus, three users were separated from the training process to produce the concatenated words for the development and test sets. For each user, the handwritten words were generated by concatenating random character instances from the user's data to form a single stroke. A summary of this corpus is shown in Table 2.5. Finally, the generated handwritten words were decoded using the proposed systems with *iAtros* [Luján-Mares et al., 2008][13] decoder. Development 3-gram perplexities are 205 for Spanish and 242 for English while test perplexities are 226 and 366, respectively. Examples of generated words are shown in Figure 2.12.

	English		Spanish	
	dev	test	dev	test
Complete sentences	510	434	576	414
Erroneous running words	2726	2248	2398	2102
Perplexity of erroneous words	242	336	205	226
Number of erroneous segments	896	1130	941	1268
Average length of erroneous segments	2.12	2.35	2.18	2.39

Table 2.5: Basic statistics of the Xerox on-line HTR corpus for English and Spanish.

[13] http://prhlt.iti.upv.es/w/iatros

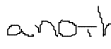
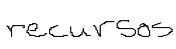
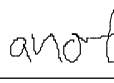
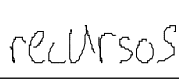
	English *another*	Spanish *recursos*
User 1		
User 2		
User 3		

Figure 2.12: Examples of pen strokes from the UNIPEN database used for the simulation of HTR for English and Spanish. The words were obtained by concatenating random character instances from the corresponding user.

2.3.2 Speech interaction for IMT

The data set for speech interaction consists of utterances of fragments of target-language (Spanish) sentences, extracted from the test part of the original Xerox corpus [Vidal et al., 2006]. These utterances are used as a test set to simulate real interactions of the IMT system with human translators. All the speech data was acquired using high quality microphones and 16KHz sampling frequency. A summary of relevant features of this corpus is shown in Table 2.6.

Text	Original complete sentences	128
	Different sentence fragments uttered	485
	Average prefix length	4.5
	Running words	1,138
	Running characters	7,320
Speech	Number of speakers	10
	Number of utterances	5,337
	Running words	13,998

Table 2.6: Spanish speech test utterances (from the Xerox corpus)

The set of test utterances was obtained as follows. First a subset of 128 sentence pairs was selected from the text partition of the Xerox text corpus. For the target (Spanish) sentence of each of these pairs, several segmentations into prefixes and suffixes were randomly performed and, for each generated suffix, a set of prefixes was randomly derived. All the prefixes of suffixes generated in this way constitute the set of sentence fragments uttered by several speakers. In order to approach real IMT user interactions as much as possible, this generation process was performed in such a way that the lengths of the generated fragments were similar to the lengths of accepted parts of system suggestions

observed in text-only experiments with a real IMT system applied to the original set of 128 sentence pairs.

The speech models are HMM with three states, with left-to-right topology with loops and 128 Gaussians per state. Each speech model represents a phonetically context-independent unit (monophone). These models were estimated using the data of the Albayzin Spanish speech corpus [Moreno et al., 1993], a phonetically balanced corpus with $42k$ running words (4 hours of speech) and 164 different speakers. Lexical models consist of stochastic finite-state machines, representing all the possible concatenations of individual characters or phonemes to compose the word. Finally, Kneser and Ney [1995] back-off smoothed 3-gram models were used as language models in all the experiments.

2.3.3 Dictation of historical documents

This corpus comprises a series of dictations of handwritten sentences of historical documents. More specifically, the experiments were carried out by dictating parts of the "Cristo-Salvador" corpus. 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.

In order to assess the speech dictation systems five different users dictated a selected page from the *Cristo Salvador* corpus, line by line. That page was selected on the basis that the average WER for this page was closest to the average WER for the whole test set. It resulted in a test data-set composed by 120 dictated lines. Some basic details are shown in Table 2.7. The acoustic HMMs needed in the speech recognition system were trained on the same corpus than for the speech-enabled IMT, the *Albayzin* corpus (cf. Section 2.3.2). The baseline language model for text lines is a 2-grams with Kneser and Ney [1995] back-off smoothing directly estimated from the training transcriptions of the text line images.

Speakers	5
Sentences	120
Running words	222
Running characters	1,239
Length (seconds)	454

Table 2.7: Basic statistics of the speech dictation test.

2.4 Evaluation metrics

The metrics used to evaluate interactive systems are based on the human effort needed to produce a correct output or reference. These metrics often

come from the normalization of the loss function at stake. Therefore, the post-editing effort in non-interactive systems has traditionally been measured by the *word error rate* (WER), as the ratio between the number of editions (substitutions, deletions and insertions) necessary to transform the hypothesis into the reference, and the number of words in the reference. A specific case is the Hamming distance, where only substitutions are needed since there is a one-to-one correspondence between the input and the output. Accordingly, in the case of classification or sequence labeling problems we will only account for substitutions. We will call this metric *classification error rate* (CER).

Arguably, the edit distance is a simplistic approach to assess PE effort. On the one hand, it is optimistic regarding the number of operations to make, since a human will hardly ever perform the operations with the *minimum* number of operations, especially in complex problems. On the other hand, the edit distance assigns the same cost to all the operations, regardless of the complexity of the problem and the cognitive effort needed to perform them. On the positive side, the edit distance provides an automatic and intuitive measure. Consequently, it has been widely adopted for many NLP tasks as the PE cost.

With respect to interactive measures, the *word stroke ratio* (WSR) measures the human error in correcting in a (passive) interactive scenario following a sequential order. It can be computed as the ratio between the number of interactions (corrected words) in the interactive system and the number of words in the reference. On the other hand, in the active interactive case we will measure the number of supervisions made, the number of corrections, and the residual error after the corrections.

Analogously, in ISP systems, the cost of interactively correcting a system output can be computed as the number of corrections (substitutions) needed to obtain the reference. Note that this is not equivalent to the Hamming distance since the part on the right of the output may change after each user correction. In this case, the cognitive effort is also neglected and the substitution cost is the same for all corrections. Besides, system suggestions may influence human corrections, since a good proposal could change a user's opinion regarding what the correct solution is. In this sense, using a unique reference can be deemed as a pessimistic approach in problems with multiple correct solutions. Furthermore, ISP systems may work at character level (*key stroke ratio* (KSR)), as opposed to word level. In character level ISP, the suffixes are computed on each user key press, instead of predicting a suffix after signaling the end of a word by pressing the space key. It is not clear whether KSR or WSR correlate better with actual human effort. However, during this thesis we will assume that KSR is, somehow, more related to the mechanical effort of typing, whereas WSR is more related to the cognitive effort. Hence, word edit distance seems a more intuitive measure. Thus, we will use word level measures throughout all this thesis. Notwithstanding some of the shortcomings presented, WSR can be considered to be a reasonably valid approximation to the human effort in

an interaction scenario.

The performance of the on-line HTR system was assessed with CER for isolated word recognition whereas WER was used in the continuous recognition experiments. In addition, in order to evaluate the improvements in the language modeling capabilities of different context-aware language modeling techniques, perplexity was employed [Rosenfeld, 2000]. Perplexity, measured for a text with respect to a language model, is a function of the likelihood of that text being produced by repeated application of the model. Similarly, *oracle* WER (OWER) is the best WER that can be obtained from the word graph resulting from the decoding process. OWER was used to evaluate lattice quality, since we expect that applying contextual information *before* the decoding of the input modality will result in a better lattice quality.

Finally, significance of our results, where used, were assessed by the *paired bootstrap resampling* method, described in [Bisani and Ney, 2004]. This technique compares two systems and finds out whether one of them significantly outperforms the other one.

Bibliography

J. Andrés-Ferrer and A. Juan. A phrase-based hidden semi-Markov approach to machine translation. In *Proc. of the European Association for Machine Translation (EAMT'09)*, p. 168–175, 2009.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J. Vilar. Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1):3–28, 2009.

L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

R. Bellman. *Dynamic Programming*. Dover Books on Mathematics. Dover, 2003.

M. Bisani and H. Ney. Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, p. 409–412, 2004.

P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2): 263–311, 1993.

S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. of the 34th annual meeting on Association for Computational Linguistics (ACL'96)*, p. 310–318, 1996.

J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

J. Esteban, J. Lorenzo, A. Valderrábanos, and G. Lapalme. TransType2: an innovative computer-assisted translation system. In *Proc. of the Annual Meeting of the ACL on Interactive poster and demonstration sessions (ACL'01)*, p. 94–97, 2004.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems (NIPS'04)*, p. 513–520, 2004.

I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Proc. of International Conference on Pattern Recognition (ICPR'94)*, p. 29–33, 1994.

F. Jelinek. *Statistical methods for speech recognition*. MIT Press, 1997.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35 (3):400–401, 1987.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, volume 1, p. 181–184, 1995.

P. Koehn. A process study of computer-aided translation. *Machine Translation*, 23(4): 241–263, 2009.

P. Koehn. *Statistical machine translation*, volume 11. Cambridge University Press, 2010.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*, p. 48–54, 2003.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (ACL'07)*, p. 177–180, 2007.

P. Liu and F. K. Soong. Word graph based speech rcognition error correction by handwriting input. In *Proc. of the 8th international conference on Multimodal interfaces (ICMI'06)*, p. 339–346, 2006.

M. Luján-Mares, V. Tamarit, V. Alabau, C. D. Martínez-Hinarejos, M. Pastor-i Gadea, A. Sanchis, and A. H. Toselli. iATROS: A speech and handwritting recognition system. In *Proc. of the V Jornadas en Tecnologías del Habla (VJTH'2008)*, p. 75–78, 2008.

C. Lundsteen, J. Phillip, and E. Granum. Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.

C. Martínez, H. García, A. Juan, and F. Casacuberta. Chromosome classification using continuous hidden markov models. *Pattern Recognition and Image Analysis*, p. 494–501, 2003.

C. Martínez, A. Juan, and F. Casacuberta. Iterative Contextual Recurrent Classification of Chromosomes. *Neural Processing Letters*, 26(3):159–175, 2007.

M. Mohri. Weighted automata algorithms. In *Handbook of weighted automata*, p. 213–254. 2009.

A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J. B. Marino, and C. Nadeu. Albayzin speech database: Design of the phonetic corpus. In *Proc. of the Third European Conference on Speech Communication and Technology (eurospeech'93)*, p. 175–178, 1993.

D. Ortiz, I. García-Varea, and F. Casacuberta. Thot: a Toolkit To Train Phrase-based Statistical Translation Models. In *Proc. of the Machine Translation Summit X (MTSUMMIT'05)*, p. 141–148, 2005.

S. Ortmanns, H. Ney, and X. Aubert. A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition. *Computer Speech and Language*, 11(1):43–72, 1997.

D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. A. Lund, and M. A. Przybocki. 1993 benchmark tests for the ARPA spoken language program. In *Proc. of the workshop on Human Language Technology (HLT'94)*, p. 49–74, 1994.

M. Pastor, A. Toselli, and E. Vidal. Writing speed normalization for on-line handwritten text recognition. In *Proc. of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, p. 1131–1135, 2005.

L. Rabiner. A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. *Proc. of the IEEE*, 77:257–286, 1989.

G. Ritter, M. Gallegos, and K. Gaggermeier. Automatic context-sensitive karyotyping of human chromosomes based on elliptically symmetric statistical distributions. *Pattern Recognition*, 28(6):823–831, 1995.

V. Romero, A. Toselli, L. Rodríguez, and E. Vidal. Computer Assisted Transcription for Ancient Text Images. In *Proc. of the International Conference on Image Analysis and Recognition (ICIAR 2007)*, p. 1182–1193, 2007.

R. Rosenfeld. Two Decades Of Statistical Language Modeling: Where Do We Go From Here? In *Proc. of the IEEE*, volume 88, p. 1270–1278, 2000.

E. Schröck, S. du Manoir, T. Veldman, B. Schoell, J. Wienberg, M. A. Ferguson-Smith, Y. Ning, D. H. Ledbetter, I. Bar-Am, D. Soenksen, Y. Garini, and T. Ried. Multicolor spectral karyotyping of human chromosomes. *Science*, 273(5274):494–7, 1996.

J. Tomás and F. Casacuberta. Monotone statistical translation using word groups. In *Proc. of the Machine Translation Summit VIII (MT SUMMIT'01)*, p. 357–361, 2001.

A. Toselli, E. Vidal, and F. Casacuberta, editors. *Multimodal Interactive Pattern Recognition and Applications*. Springer, 2011.

A. H. Toselli, A. Juan, J. González, I. Salvador, E. Vidal, F. Casacuberta, D. Keysers, and H. Ney. Integrated handwriting recognition and interpretation using finite-state models. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(4): 519–539, 2004.

A. H. Toselli, V. Romero, M. Pastor, and E. Vidal. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.

N. Ueffing, F. Och, and H. Ney. Generation of word graphs in statistical machine translation. In *Proc. of the conference on Empirical methods in natural language processing (EMNLP'02)*, p. 156–163, 2002.

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

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

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, 2006.

F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.

S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, 2006.

R. Zens, F. Och, and H. Ney. Phrase-based statistical machine translation. In *Proc. of the 25th Annual German Conference on Artificial Intelligence (KI'02)*, p. 35–56, 2002.

Chapter 3

Passive Interactive Structured Prediction

Chapter Outline

3.1 Introduction

As we have seen in Chapter 1, the *interactive structured prediction* (ISP) framework (also known as *interactive pattern recognition* [Toselli et al., 2011]) was introduced in [Vidal et al., 2007] to reduce the cost of correcting the automatically generated output. In ISP, the user is introduced in the core of a SP system so that the system and the user can interact with each other to minimize the effort required to produce a satisfactory output (Figure 3.1 represents the ISP interaction scheme). In ISP, an input $\boldsymbol{x}$ is given to the system, which outputs a possible hypothesis $\hat{\boldsymbol{y}}$. Then, the user analyzes $\hat{\boldsymbol{y}}$ and provides feedback $\boldsymbol{f}$ regarding some of the errors committed. Now, the system can benefit from the feedback to propose a new improved hypothesis. This process is repeated until the user finds a satisfactory solution, $\boldsymbol{r}$, and the process ends.

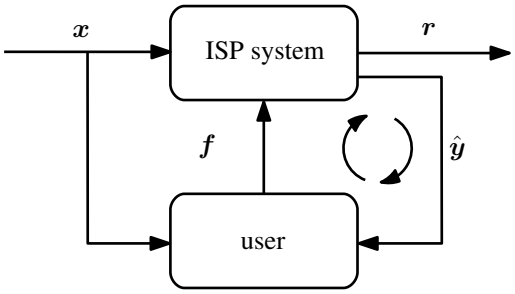


Figure 3.1: Diagram of an *passive interactive structured prediction* process. The system processes the input $\boldsymbol{x}$ to produce an initial output $\hat{\boldsymbol{y}}$. Then, the user analyses the output and proposes a correction by some feedback $\boldsymbol{f}$. Now, the system proposes a new hypothesis $\hat{\boldsymbol{y}}$. This process is repeated until the desired solution $\boldsymbol{r}$ is obtained.

Traditionally, ISP systems are designed following a decision rule to minimize output errors. However, this is not optimal for ISP since the decision rule should be formalized in terms of minimizing user interactions. Indeed, this fact was proved in [Oncina, 2009], where an alternative strategy is applied to a specific case of ISP (i.e., text prediction). Inspired by that work, here we provide an optimal decision rule for ISP which covers a broader range of common ISP problems in which the output depends on a structured input $\boldsymbol{x}$. This strategy is analyzed from a theoretical perspective and a practical decoding algorithm is developed to be used straightforwardly in many ISP tasks. In addition, we propose theoretically and empirically that the algorithm that has been used for ISP until now is a good approximation to the optimum for ISP.

3.2 Sequential Interactive Structured Prediction

Sequential interactive structured prediction (SISP) is a specific case of ISP where

the user validates and/or corrects the system output in sequential order (typically left-to-right). This is especially interesting for many *natural language processing* (NLP) tasks, since humans usually listen, read, write, and talk in sequential order. Other non-sequential protocols may seem more natural in some scenarios, for example in interactive predictive parsing. Nonetheless, even in this case, the SISP protocol has been proven to be useful [Sánchez-Sáez et al., 2009].

Let $\boldsymbol{y}^{(i)}$ be a sequence of labels (e.g., words) that represents a hypothesis at iteration (i). We can define $\boldsymbol{y}^{(i)}$ as the concatenation[1] of a correct prefix $\boldsymbol{y}_p^{(i)}$, that matches a prefix of the solution $\boldsymbol{r}$, and a suffix hypothesis $\boldsymbol{y}_s^{(i)}$, so that $\boldsymbol{y}^{(i)} = \boldsymbol{y}_p^{(i)} \cdot \boldsymbol{y}_s^{(i)}$. Then, the protocol that rules SISP to obtain $\boldsymbol{r}$ can be formulated in the following steps:

0. Initially $(i = 0)$, the correct prefix is the empty string, $\boldsymbol{y}_p^{(0)} = \lambda$, and the system proposes a complete hypothesis $\hat{\boldsymbol{y}}_s^{(0)}$.
1. At iteration $(i \geq 1)$, the user finds the longest prefix $\boldsymbol{a}^{(i)}$ of $\hat{\boldsymbol{y}}_s^{(i-1)}$ that is error-free and corrects the first error in the suffix, which, let us assume, is at position k, with r_k.
2. A new extended prefix $\boldsymbol{y}_p^{(i)}$ is produced as a concatenation of the correct prefix, the new error-free segment of the suffix and the new introduced word $\boldsymbol{y}_p^{(i-1)} \cdot \boldsymbol{a}^{(i)} \cdot r_k$.
3. Then, the system proposes a suffix hypothesis $\hat{\boldsymbol{y}}_s^{(i)}$ that follows the prefix $\boldsymbol{y}_p^{(i)}$ established in the previous step.
4. Steps 1, 2 and 3 are iterated until, at some iteration $i = I$ with $I \leq |\boldsymbol{r}|$, a correct solution is obtained, $\hat{\boldsymbol{y}}^{(I)} = \boldsymbol{r}$.

Figure 3.2 shows an example of this SISP protocol for MT. Initially, the system starts with an empty prefix and a full hypothesis is proposed. The user finds the first error, '*cannot*', and amends it with the correct word '*is*'. Since the user validates sequentially from left to right, the system assumes that the prefix $\boldsymbol{y}_p^{(1)}$ '*if any feature is*' is correct. Based on this validated prefix, the system produces a new suffix $\boldsymbol{y}_s^{(1)}$, in which the words '*be found on*' have been automatically corrected by '*not available at*'. Similarly, at iteration 2, by introducing r_7 ='*in*' the system corrects '*web*' at the end of the sentence. Finally, the system suggests a new suffix $\boldsymbol{y}_s^{(2)}$ that is correct and the user ends the process. Note that in a PE system, the user would have needed to make five corrections whereas just two corrections are needed in SIPS.

[1] The symbol · will be used to denote the concatenation of two or more variables.

SOURCE ($\boldsymbol{x}$): si alguna función no se encuentra disponible en su red
REFERENCE ($\boldsymbol{r}$): if any feature is not available in your network

$i = 0$	$\boldsymbol{y}_p^{(0)}$	
	$\hat{\boldsymbol{y}}_s^{(0)}$	if any feature cannot be found on your web
	$\hat{\boldsymbol{y}}^{(0)}$	if any feature cannot be found on your web
$i = 1$	$\boldsymbol{a}^{(1)}$	*if any feature*
	r_4	**is**
	$\boldsymbol{y}_p^{(1)}$	if any feature is
	$\hat{\boldsymbol{y}}_s^{(1)}$	not available at your web
	$\hat{\boldsymbol{y}}^{(1)}$	if any feature is not available at your web
$i = 2$	$\boldsymbol{a}^{(2)}$	*not available*
	r_7	**in**
	$\boldsymbol{y}_p^{(2)}$	if any feature is not available in
	$\hat{\boldsymbol{y}}_s^{(2)}$	your network
	$\hat{\boldsymbol{y}}^{(2)}$	if any feature is not available in your network
$I = 2$	$\boldsymbol{y}^{(2)} = \boldsymbol{r}$	if any feature is not available in your network

Figure 3.2: Example of a SISP session for a MT task from Spanish to English. The source sentence is the input $\boldsymbol{x}$ while the reference $\boldsymbol{r}$ is the result that the user has in mind. At each iteration (i), $\hat{\boldsymbol{y}}_s^{(i)}$ is the suffix proposed by the system. $\boldsymbol{a}^{(i)}$ (in *italics*) is the longest correct prefix of $\hat{\boldsymbol{y}}_s^{(i-1)}$. Finally, r_k (in **boldface**) is the word introduced by the user to amend the error, which results in a new validated prefix $\boldsymbol{y}_p^{(i)}$. Note that only two user corrections have been needed to produce a correct solution whereas five edition operations would have been necessary with PE.

3.3 Optimal Decision Rule for SISP

Ideally, when building a SISP system, we would like to devise a system which allows the human expert to amend the system output with less effort. As we explained in Chapter 1, following the *minimum classification error* (MCE) [Duda et al., 2001], the optimum decision rule is the one that minimizes the average probability of loss (conditional risk) over a probability distribution $\Pr(\cdot)$. In classification problems, the human effort can be approximated by a cost 0 if the output is correct and a cost 1 if the user has to amend an erroneously classified sample by assigning the correct label. This function is known as the *zero-one* loss function and leads to the *maximum-a-posteriori* (MAP) decision rule [Duda et al., 2001]:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \tag{3.1}$$

Typically, SISP systems have used an extension of MAP which has been successfully applied to several NLP tasks [Toselli et al., 2011], namely interactive machine translation [Barrachina et al., 2009], interactive transcription of text

images [Toselli et al., 2010], interactive predictive parsing [Sánchez-Sáez et al., 2009], interactive speech transcription [Rodríguez et al., 2007] and interactive text generation [Rodríguez et al., 2010]. Thus, at steps 0 and 3 of the SIPS protocol, a suffix hypothesis $\hat{\boldsymbol{y}}_s^{(i)}$ that continues a validated prefix $\boldsymbol{y}_p^{(i)}$ is generated using the following equation:

$$\hat{\boldsymbol{y}}_s^{(i)} = \arg\max_{\boldsymbol{y}_s} \Pr(\boldsymbol{y}_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \tag{3.2}$$

where the output in the interaction (i) is $\hat{\boldsymbol{y}}^{(i)} = \boldsymbol{y}_p^{(i)} \cdot \hat{\boldsymbol{y}}_s^{(i)}$. Nevertheless, the MAP rule seeks solutions with zero errors, whereas we would prefer a hypothesis that minimizes the number of human corrections.

3.3.1 The Cost of Interactively Correcting the Output

In Section 2.4, we introduced *word stroke ratio* (WSR) as the metric for SIPS. We also explained that *key stroke ratio* (KSR) could be also a valid metric, although WSR seemed more appropriate to represent user's cognitive effort. Both metrics express the ratio of corrections with respect to the total number of label. However, in WSR the corrections will represent *word corrections*, whereas in KSR they will represent *character corrections*. Mathematically, we can use the same expression for both cases to define the cost of interaction by counting the number of corrections.

Definition 3.1. *The SISP protocol has an associated cost function* $C(\boldsymbol{y}^{(i)}, j, \boldsymbol{r})$ *that computes the cost of sequentially producing a reference* $\boldsymbol{r}$ *from the label at position* j *of the hypothesis* $\boldsymbol{y}^{(i)}$ *at iteration* (i)*. Using the abbreviated notation,* $C_j^{(i)}$*, the cost can be described as:*

$$C_j^{(i)} = \bar{\delta}_j^{(i)} \left(1 + C_{j+1}^{(i+1)}\right) + \delta_j^{(i)} C_{j+1}^{(i)} \tag{3.3}$$

where $\delta_l^{(i)} = \delta(y_l^{(i)}, r_l)$ *is a* Kronecker delta *function that is* 1 *if* $y_l^{(i)} = r_l$ *and* 0 *otherwise, whereas* $\bar{\delta}_l^{(i)}$ *is the negation of* $\delta_l^{(i)}$*.*

It is worth of noting that Equation (3.3) is different from the cost defined by the Hamming function for post-editing sequence labeling problems (cf. Section 1.4.1). Although both costs count the number of corrections by performing substitutions, in Equation (3.3) the system is dynamic, meaning that the hypotheses and problem constraints change over time, as a product of user's interaction. Conversely, in post-editing of sequence labeling problems the system is static.

As a remainder, we should remark that we will denote with y_k the k-th element in $\boldsymbol{y}$ and with $y_{j..k-1}^{(i)}$ the substring from j to $k-1$ of $\boldsymbol{y}^{(i)}$.

Proposition 3.1. *Let* $\Pr(\boldsymbol{y}_s|\boldsymbol{x}, \boldsymbol{y}_p^{(i)})$ *be a posterior probability over the suffixes that continue* $\boldsymbol{y}_p^{(i)} = (y_1^{(i)}, \ldots, y_{j-1}^{(i)})$. *A suffix* $\hat{\boldsymbol{y}}_s^{(i)} = (\hat{y}_j^{(i)}, \ldots, \hat{y}_k^{(i)}, \ldots, \hat{y}_{\hat{I}}^{(i)})$ *with the last symbol at position* $\hat{I}$*, which minimizes the conditional risk of the number of interactions, can be obtained following this decision rule:*

$$\hat{y}_k^{(i)} = \arg\max_{y_k} \sum_{\boldsymbol{y}'_s} \Pr(y_k \cdot \boldsymbol{y}'_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot \hat{y}_{j..k-1}^{(i)})$$
$$\text{for } k = j \ldots \hat{I} \wedge \hat{y}_{\hat{I}+1}^{(i)} = \$ \tag{3.4}$$

where $\boldsymbol{y}'_s$ *is a possible suffix for* $\boldsymbol{y}_p^{(i)} \cdot \hat{y}_{j..k-1}^{(i)} \cdot y_k$ *and* \$ *is a special symbol that means the end of the hypothesis.*

Proof. Following the MCE approach, an optimum algorithm for SIPS is one that minimizes the conditional expected value ($\mathrm{E}((|\cdot|))$) of $C_j^{(i)}$:

$$(\hat{y}_j^{(i)}, \ldots, \hat{y}_{\hat{I}}^{(i)}) = \arg\min_{(y_j, \ldots, y_I)} \min_I \mathrm{E}\left(C_j^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)}\right) \tag{3.5}$$

As the expected value is a linear operator, and after simple transformations, Equation (3.5) becomes:

$$\begin{aligned}
\mathrm{E}\left(C_j^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)}\right) &= \mathrm{E}\left(\bar{\delta}_j^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)}\right) + \mathrm{E}\left(\bar{\delta}_j^{(i)} C_{j+1}^{(i+1)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)}\right) \\
&\quad + \mathrm{E}\left(\delta_j^{(i)} C_{j+1}^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)}\right) \\
&= 1 - \sum_{\boldsymbol{y}'_s} \Pr(y_j^{(i)} \cdot \boldsymbol{y}'_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) + \sum_{w \neq y_j^{(i)}} \mathrm{E}\left(C_{j+1}^{(i+1)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot w\right) \\
&\quad + \mathrm{E}\left(C_{j+1}^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot y_j^{(i)}\right)
\end{aligned} \tag{3.6}$$

First, note that the sum of the expected values of $C_{j+1}^{(i)}$ and $C_{j+1}^{(i+1)}$ in Equation (3.6) covers all possible suffixes of $\boldsymbol{y}_p^{(i)}$. Hence, this sum is constant for every possible value of y_j and the minimization can be done independently. Then,

$$\begin{aligned}
\hat{y}_j^{(i)} &= \arg\min_{y_j} \left(1 - \sum_{\boldsymbol{y}'_s} \Pr(y_j \cdot \boldsymbol{y}'_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)})\right) \\
&= \arg\max_{y_j} \sum_{\boldsymbol{y}'_s} \Pr(y_j \cdot \boldsymbol{y}'_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)})
\end{aligned} \tag{3.7}$$

Consequently, $\hat{y}_j^{(i)}$ must form part of the optimum hypothesis. The minimization for subsequent elements can be rewritten as

$$(\hat{y}_j^{(i)}, \ldots, \hat{y}_{\hat{I}}^{(i)}) = \underset{(y_j,\ldots,y_I):y_j=\hat{y}_j^{(i)}}{\arg\min} \min_I \mathrm{E}\left(C_j^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)}\right) \tag{3.8}$$

Since all but the last term in Equation (3.6) are constant now that $\hat{y}_j^{(i)}$ has been fixed,

$$(\hat{y}_j^{(i)}, \ldots, \hat{y}_{\hat{I}}^{(i)}) = \underset{(y_j,\ldots,y_I):y_j=\hat{y}_j^{(i)}}{\arg\min} \min_I \mathrm{E}\left(C_{j+1}^{(i)} \middle| \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot \hat{y}_j^{(i)}\right) \tag{3.9}$$

Similarly to Equation (3.7), we can obtain $\hat{y}_{j+1}^{(i)}$. Now, by induction it is trivial to prove that, if Equation (3.9) holds for $\hat{y}_{j..k-1}^{(i)}$, it also holds for $\hat{y}_k^{(i)}$, using the same reasoning. That concludes the proof of Proposition 3.1. □

The algorithm in Proposition 3.1 works by constructing the output incrementally, by appending individual labels from left to right. The decision of appending a new label is conditioned to previous labels, but it is independent of future decisions. The idea behind this is that, if the user amends a label at position l, $\hat{y}_l^{(i)}$, all of the following labels in $\hat{\boldsymbol{y}}^{(i)}$ (i.e., $\hat{y}_{l+1..\hat{I}}^{(i)}$) will be discarded in favor of a new suffix, $\hat{\boldsymbol{y}}_s^{(i+1)}$. Hence, they are not relevant for the decision process.

It must be noted that the proof assumes that the process is stationary, i.e., the posterior probability does not change during the computation of the decision rule. Therefore, if an *incremental learning* algorithm is used after each user interaction, then $\Pr(\cdot)$ may change. As a result, the sum of the expected values of $C_{j+1}^{(i)}$ and $C_{j+1}^{(i+1)}$ in Equation (3.6) cannot be considered a constant since the probability distribution of the expected value can be different, and thus, the expected value itself. However, if *on-line learning* were used after the sentence had been completely corrected, then Proposition 3.1 would still be an optimum algorithm.

3.3.2 Relation with the MAP Decision Rule

It has been mentioned that the MAP decision rule (Equation (3.2)) has been extensively used for SISP. Although Equation (3.2) is known to minimize the *zero-one* loss function for hypothesis suffixes, it would be interesting to analyze how it behaves with respect to Equation (3.4).

Proposition 3.2. *The MAP decision rule is equivalent to a maximum approximation to the optimal decision rule for SISP,*

$$\hat{y}_k^{(i)} = \arg\max_{y_k} \max_{\boldsymbol{y}'_s} \Pr(y_k \cdot \boldsymbol{y}'_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot \hat{y}_{j..k-1}^{(i)})$$
$$\text{for } k = j \dots \hat{I} \wedge \hat{y}_{\hat{I}+1}^{(i)} = \$ \tag{3.10}$$

Proof. For $k = j$, Equation (3.2) and Equation (3.10) are obviously equivalent. Obtaining y_j from Equation (3.2),

$$\hat{y}_j^{(i)} \overset{(3.2)}{=} \arg\max_{y_j} \max_{y_{j+1..I}} \Pr(y_{j..I} | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = \arg\max_{y_j} \max_{y_{j+1..I}} \Pr(y_j \cdot y_{j+1..I} | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \tag{3.11}$$

for $\boldsymbol{y}_s = y_{j+1..I}$.

Then, by induction, if both decision rules are equivalent for $\hat{y}_{j..k-1}^{(i)}$, then they are also equivalent for $\hat{y}_k^{(i)}$,

$$\begin{aligned} \hat{y}_k^{(i)} &\overset{(3.2)}{=} \arg\max_{y_k} \max_{y_{j..k-1}, y_{k+1..I}} \Pr(y_{j..I} | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \\ &= \arg\max_{y_k} \max_{y_{j..k-1}, y_{k+1..I}} \Pr(y_{j..k-1} | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \Pr(y_k \cdot y_{k+1..I} | \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot y_{j..k-1}) \end{aligned} \tag{3.12}$$

Since $\hat{y}_{j..k-1}^{(i)}$ are known to be the optimum values, the first term of the product is constant in Equation (3.12) finally reaching

$$\hat{y}_k^{(i)} \overset{(3.2)}{=} \arg\max_{y_k} \max_{y_{k+1..I}} \Pr(y_k \cdot y_{k+1..I} | \boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot \hat{y}_{j..k-1}) \tag{3.13}$$

for $\boldsymbol{y}_s = y_{k+1..I}$. Therefore, both decision rules are equivalent. □

Proposition 3.2 has two main implications. Firstly, it provides a formalism for the traditional MAP approach as it can be seen as a maximum approximation to the optimum. That is especially convenient for models where Equation (3.4) cannot be computed efficiently (exponential number of suffixes). Secondly, on non-smooth probability distributions where the mass probability is concentrated around the maximum, MAP performs almost as accurately as the optimum algorithm.

Strictly speaking, MAP is not an admissible rule under a decision theory perspective. We say that a decision rule is inadmissible if there is another decision rule that dominates it, i.e. there is a rule for which the risk is always better or equal for a given loss function.

Corollary 3.1. *The MAP decision rule is an inadmissible decision rule for the interactive cost function.*

Proof. It is simple to prove that optimum decision rule always dominates the MAP rule, since the latter is a maximum approximation of a sum of positive elements, an thus always equal or better. □

Nonetheless, MAP continues to be a pragmatic and interesting approach to SISP. The relation of MAP and the decision rule to minimize the edit distance was studied in [Schlüter et al., 2005, 2010, 2012], for non-interactive systems. In those papers it was shown that both decision rules coincide under certain conditions when the cost function is an integer-valued metric. In practice, Schlüter et al. [2010] experiments showed that, for some ASR tasks, between 73.5% and 95.8% of the sentences resulted in the same hypothesis for an optimum and MAP decision rules. Although the cost function for SISP is not a metric by itself, some of these conditions could also hold for SISP problems.

Corollary 3.2. *(Loss-Independence of the Bayes Decision Rule for Large Posterior Probability [Schlüter et al., 2005]) Assume a maximum posterior probability* $\geq \frac{1}{2}$ *and a loss function defined by* $C_j^{(i)}$*. Then the posterior maximizing class also maximizes the Bayes risk.*

Proof. This proposition was defined for metric loss functions but it is also true for SISP. First, the sum of the posteriors for all possible suffixes must be 1,

$$\sum_{y_j} \sum_{\boldsymbol{y}'_s} \Pr(y_j \cdot \boldsymbol{y}'_s | \boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = 1 \tag{3.14}$$

Thus, if ${y_j}^+$ is the hypothesis with maximum posterior probability, and this probability is $\geq \frac{1}{2}$, then the remaining probability mass is $\leq \frac{1}{2}$. Hence, there cannot be a $y_j \neq {y_j}^+$ for which the sum in Proposition 3.1 is $\geq \frac{1}{2}$. By induction, it is trivial to prove that this is also true $\forall k : j < j \leq I$. □

In addition, Schlüter et al. [2010] established the Hamming risk as an upper bound of the risk to the edit distance risk. The intuition points out that this will also be true for SISP, although we do not present a prove in this thesis.

3.3.3 Relation with the *greedy* algorithm

The proof provided for Proposition 3.1 is inspired in previous work by Oncina [2009], who reached a similar algorithm. In fact, it is possible to integrate the summation for all suffixes $\boldsymbol{y}'_s$, producing the optimum algorithm in [Oncina,

2009] where $\boldsymbol{x}$ is shown explicitly,

$$\hat{y}_k^{(i)} = \arg\max_{y_k} \Pr(y_k|\boldsymbol{x}, \boldsymbol{y}_p^{(i)} \cdot \hat{y}_{j..k-1}^{(i)}) \quad \text{for } k = j \dots \hat{I} \wedge \hat{y}_{\hat{I}+1}^{(i)} = \$ \tag{3.15}$$

However, Oncina [2009] deduced that the decision of a label in the output did not depend on the rest of the labels that followed it. As a result, the optimum decision rule was deemed as greedy in the sense that decisions could be taken locally. That is, the probability in Equation (3.15) models only the next label, whereas in Equation (3.4) it models whole suffixes. Probably, this deduction was due to the fact that, in his algorithm, Oncina [2009] ignored $\boldsymbol{x}$ from the formulation since $\boldsymbol{x}$ would not change between interactions, and thus, this particular problem was hindered. Unfortunately, a greedy algorithm is not a solution when dealing with problems with an structured input, especially if the original problem is best solved by an algorithm that takes global decisions. In those cases, such as the problems we defined in Chapter 2, the directly modeling the "local" posterior probability in Equation (3.15) is still an unresolved problem, especially those with latent variables. There exist approaches using incremental models [Daumé III et al., 2009; Maes et al., 2009] (as opposite to global models), that could be used to estimate such posterior probabilities directly. However, they are typically used for sequence labeling tasks, where the output is the same length as the input. Therefore, it is necessary to rely on the global models used in Equation (3.4) and perform the sum explicitly over a (potentially) exponential number of suffixes.

3.4 Practical decoding algorithm

The algorithm described in Proposition 3.1 has the difficulty that the sum over all possible suffixes must be done explicitly. In practice, this may be a major problem, since SP outputs are combinatorial by nature. Hence, to list all possible suffixes can be a hard problem. To deal with this problem, we propose a general practical algorithm for optimal decoding. The whole set of hypotheses (search space) will be represented by a word graph.

Given the input $\boldsymbol{x}$, the posterior probability for a specific edge e can be computed by summing up the posterior probabilities of all hypotheses of the word graph containing e. These posterior probabilities (here we use small p to denote models instead of true probabilities) can be efficiently computed based on the well-known *forward-backward* algorithm [Wessel et al., 2001],

$$p(e|\boldsymbol{x}) = p((y,u,v)|\boldsymbol{x}) = \frac{\Phi(u) F_{\boldsymbol{x}}(e) \Psi(v)}{\Phi(q_f)} \tag{3.16}$$

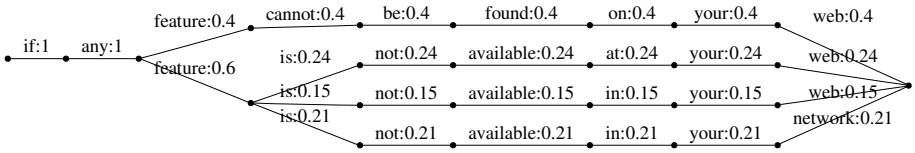


Figure 3.3: Word graph obtained as the translation of the input sentence in Figure 3.2. Edges show the hypothesized word and its posterior probability.

where the forward score $\Phi(u)$ for node u is the sum of all possible paths from the initial node q_i to u. Similarly, the backward score $\Psi(v)$ for node v is the sum of all possible paths from v to the final node q_f. Figure 5.3 shows a (pruned) word graph obtained as the result of the translation of the input sentence for the example in Figure 3.2, after the word posterior probabilities have been computed.

Now, we can conveniently introduce the dependency on the prefix $\boldsymbol{y}_p^{(i)}$ in Equation (3.16):

$$p(e|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = p((y,u,v)|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = \frac{\Phi_{\boldsymbol{y}_p^{(i)}}(u) F_{\boldsymbol{x}}(e)\, \Psi(v)}{Z_{\boldsymbol{y}_p^{(i)}}} \tag{3.17}$$

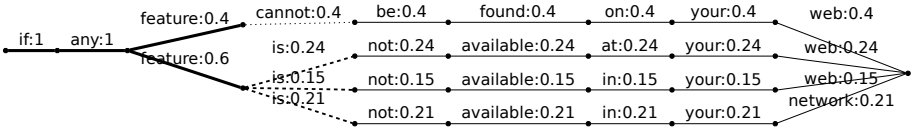
Note that the prefix dependency affects the forward score $\Phi_{\boldsymbol{y}_p^{(i)}}$. In this case, Equation (3.17) is restricted to the sum of all paths from the initial node q_i to u for which the sequence of labels matches the prefix $\boldsymbol{y}_p^{(i)}$. Also, the normalization factor $Z_{\boldsymbol{y}_p^{(i)}}$ now only takes into account the mass probability of all the paths that have $\boldsymbol{y}_p^{(i)}$ as a prefix.

Then, Equation (3.4) can be easily computed by marginalizing over all of the edges with the word y that follow $\boldsymbol{y}_p^{(i)}$:

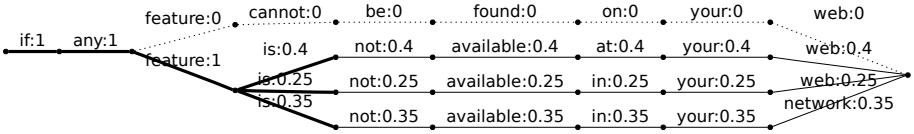
$$\sum_{\boldsymbol{y}'_s} \Pr(y \cdot \boldsymbol{y}'_s|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \simeq p(y|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = \sum_u \sum_v p((y,u,v)|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) \tag{3.18}$$

Figure 3.4 exemplifies how the state of the optimum algorithm changes when predicting the word at position $j = 4$ in iteration $(i = 0)$ for the example in Figure 3.2. Note that an error is committed by the MAP approach (Figure 3.4a) since it relies on the edge with the highest probability. In contrast, the optimum algorithm selects the set of edges with the same word whose sum is maximum, allowing the correct solution to be chosen.

Proposition 3.3. *If the word graph is a deterministic probabilistic finite-state machine (normalized as in [Sánchez et al., 2012]) then a greedy algorithm is optimum w.r.t. the cost of interaction.*



(a) State of the optimum algorithm at iteration ($i = 0$) when obtaining $j = 4$. The dashed edges suggest the paths to be chosen by the optimum algorithm, whereas the dotted edge suggests the path to be chosen by the maximum approach.



(b) State of the optimum algorithm at iteration ($i = 0$) after obtaining $j = 4$. Note that the posterior probabilities have been renormalized to contain only the paths with the prefix 'if any feature is'. Dotted edges display unreachable paths, for which the posterior probability is 0. All the paths that are compatible with the prefix are eligible candidates. Note that, based on the optimum algorithm, the correct output is produced at iteration ($i = 0$).

Figure 3.4: Word graphs for the example in Fig. 3.2 for the optimum algorithm. This figure exemplifies how the state of the algorithm changes when predicting the word at position $j = 4$ in iteration ($i = 0$). Edges show the hypothesized words and the posterior probabilities as in Eq. (3.18). Bold edges show current compatible prefixes.

Proof. First, note that if the WG is deterministic, the double summation in Equation (3.18) is made over a single element, since by definition of determinism from a node only one edge can exit with the same label. Then,

$$p(y|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = p((y,u,v)|\boldsymbol{x}, \boldsymbol{y}_p^{(i)}) = \frac{\Phi_{\boldsymbol{y}_p^{(i)}}(u) F_{\boldsymbol{x}}(e)\, \Psi(v)}{\Phi_{\boldsymbol{y}_p^{(i)}}(u) \Psi(u)} \tag{3.19}$$

$$= \frac{F_{\boldsymbol{x}}(e)\, \Psi(v)}{\Psi(u)} \tag{3.20}$$

which is the normalization solution in [Sánchez et al., 2012]. □

A practical implication of Proposition 3.3 is that SIPS decoding can be performed in linear time if the WG determinization is precomputed. Note that the general algorithm for automata determinization does not guarantee termination. However, any acyclic weighted automaton over a zero-sum-free semiring is determinizable [Mohri, 2009]. Hopefully, WGs are such kind of automata, either they work in the tropical semiring, the probability semiring, or the log semiring.

3.5 Experimentation

In order to assess under what conditions the optimal decision rule outperformed the classical approach, we first designed a simulated scenario. This

way, we could control the peakedness of the distribution. This parameter is critical for the optimum approach to outperform the classical approach since in peaky distributions, the maximum value dominates the sum, hence, both approaches become equivalent. Furthermore, we wanted to evaluate the techniques under different error rate conditions as tasks with more error rate would need a smoother distribution to allow the sum good solutions take over the maximum value. For this purpose, we obtained the references from the Wall Street Journal (WSJ) database [Pallett et al., 1994]. Then, for each reference we built a WG with 1000 hypotheses generated by introducing uniformly distributed random errors on the reference, with an ε error rate. In consequence, for each ε we generated a test set with WGs that represented a recognition process with ε errors distributed uniformly (in average). Next, we assigned a score for each hypothesis assuming that the posterior probability followed an exponential distribution, $\lambda e^{-\lambda n}$, where n is the number of hypotheses and $\lambda \geq 0$ controls the peakedness of the distribution (the bigger the peakier).

In addition to the simulated experiment, five real world NLP tasks were selected: DNI recognition, karyotype classification, machine translation, handwritten text recognition, and automatic speech recognition. These tasks were explained in Section 2.2. It must be mentioned that, as a result of the combinatorial nature of the search space for these tasks, the search algorithms used were not able to obtain the whole search space, except for the DNI recognition task. Instead, the search algorithms used heuristics to prune the space and to reduce computational costs. Consequently, the WGs contain a set of the most likely hypotheses but not all of them. In addition, as it was commented in Section 2.1, the WGs represent the unnormalized posterior distribution, which in these cases is represented in logarithmic form. Hence, we need to redefine the posterior probability of a path in a WG (Equation (2.2)) as

$$p(\boldsymbol{e} \mid \boldsymbol{x}) = \frac{\exp(F_{\boldsymbol{x}}(\boldsymbol{e})/\eta)}{\sum_{\boldsymbol{e}'} \exp(F_{\boldsymbol{x}}(\boldsymbol{e}')/\eta)} \tag{3.21}$$

where η is the posterior scale that is used to control the peakedness of the posterior distribution which is needed to balance the scores of the competing hypotheses. High values of η will make the posterior probability distribution smoother, whereas low values will make it sharper. In practice, this scaling factor adjusts the difference in probability between the first and runner up hypotheses. Finally, η are usually estimated empirically on a held data set to optimize the loss function.

It is important to note that, in the way Equation (2.5) and Equation (2.6) are defined, the result of these decision rules is invariant to the value of η. That is, η is used to modify the *difference* (sharpness) between the probabilities of the hypotheses, but it does not change their *rank* (order). Thus, the first hypothesis will be the same regardless of η. Nonetheless, η will be crucial for decision rules

where different hypotheses are summed up, as in Equation (3.18).

3.5.1 Results

We will denote the traditional MAP approach to SISP as SISP-MAP and the proposed approach as SISP-OPT. Figure 3.5 shows the evolution of the SISP-OPT decision rule as λ approaches one. It can be seen that, in this ideal scenario, when the distribution is smooth, the sum of different suffix hypotheses averages to obtain a much better result. However, as λ reaches one the distribution becomes so peaky that SISP-OPT is equal to SISP-MAP from that point on. This is especially important for the experiments with higher error rates since, as we anticipated, higher error rates require smoother distribution probabilities so that the sum of good hypotheses take over the maximum value. From Figure 3.5 we could deduce that we should always reduce the peakedness to the minimum. Unfortunately, this is an effect of the errors in the experiment being uniformly distributed. Conversely, in a real task, flat distributions would result in bad hypotheses taking over good ones, as a consequence of the noisy nature of the WG. Hence, optimization of the peakedness of the distribution is crucial for the optimum approach to succeed.

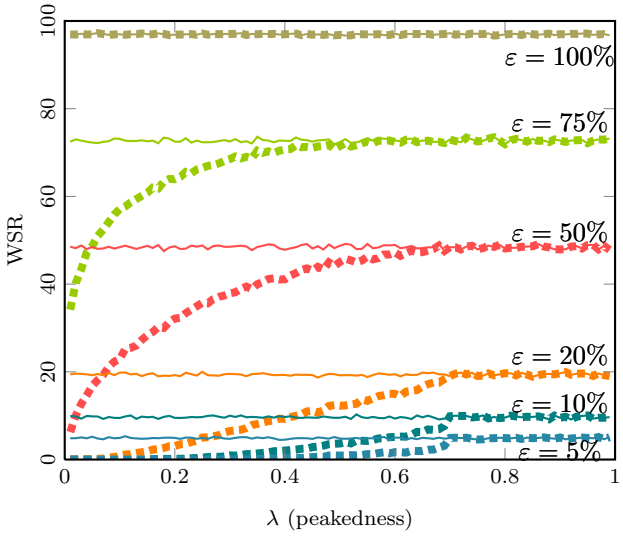


Figure 3.5: WSR as peakedness increases for different error rates ε in the simulated experiments from the WSJ corpus. The thick lines represent the WSR for SISP-OPT, whereas the thin lines represent that of SISP-MAP

The results for real tasks are presented in Table 3.1. It can be seen that the two SISP systems outperform their PE counterparts. With regard to the comparison between both SISP approaches, SISP-OPT always performs equal

to or better than SISP-MAP. For the DNI task, SISP-OPT outperforms SISP-MAP, with a *probability of improvement* (poi) of 99% using the confidence estimation technique proposed in [Bisani and Ney, 2004]. In this particular case, the WGs contain the complete search space so the decision rule is exact. With the karyotype tasks we can also see some improvements, although this time they are not statistically significant. As explained in Section 3.3.2, SISP-MAP should be close or match SISP-OPT performance for peaky probability distributions, i.e., the probability mass is concentrated around the maximum. HMMs for HTR are known to have very high 'peaks'. In fact, as there is only one possible transcription, the true probability distribution would give probability *one* to the reference and *zero* to the rest, resulting in the 'peakiest' distribution. This is reflected in the fact that SISP-MAP and SISP-OPT obtain the same exact result. SMT models are also 'peaky'. However, unlike HTR, there may exist several perfectly correct translation references for a given source sentence. This could explain why SISP-OPT manages to improve an absolute 0.1 for the Xerox English-Spanish corpus. With respect to the ASR results, statistically significant improvements can be also observed. The reason for this may be that we were able obtain more dense word lattices and, consequently, the summation in Equation (3.4) was made over a large set of suffixes.

The results for the real tasks have shown improvements on some of the tasks, while both SIPS approaches obtain the same results in the rest. This fact was already anticipated in Proposition 3.2, but the results are also supported by Schlüter et al. [2005, 2010, 2012], which concludes that the use of optimal decision rules for task-related cost functions has a limited impact.

Table 3.1: Results for real tasks. PE represents the post-editing error in a non-interactive scenario. SISP-MAP is the error of the traditional approach to SISP and SISP-OPT is the error of the optimum approach.

	DNI	Karyo	Xerox		CS	WSJ
System	–	–	en-es	es-en	es	en
PE (WER%)	1.74	3.05	24.0	27.0	33.6	15.5
SISP-MAP (WSR%)	1.15	1.41	23.3	26.3	29.6	13.2
SISP-OPT (WSR%)	0.98[a]	1.36	23.2	26.3	29.6	13.0[a]

[a] Statistically significant (poi > 99%)

3.6 Summary of contributions

In this chapter, we have presented an optimal decision rule for *sequential interactive structured prediction* (SISP) that generalizes the work on *text prediction* by Oncina [2009] to SISP problems that depend on a structured input. Our approach extends previous work by allowing full suffix prediction instead of single

symbols. This can be considered to be a relevant contribution since they represent the most frequent problems in SP. In addition, the *maximum-a-posteriori* approach was described as a maximum approximation to the optimal decision rule. Furthermore, a practical and general decoding algorithm was developed over word graphs. Experiments on different NLP tasks have shown that the MAP decision rule performs very similarly to the optimal one for non-smooth probability distributions, as was expected. However, the optimum strategy has still been able to obtain improvements.

Further work should delve into the analysis of the optimal decision rule behavior. Directly implementing the optimal decision rule instead of using word graphs would probably lead to better improvements, since the sum is made over a wider range of hypotheses. It would also be interesting to test the results on other NLP tasks. Further research should especially concentrate on finding real tasks with smooth probability distributions so that the behavior of the optimal decision rule under more favorable conditions could be analyzed. Finally, the theoretical properties of the algorithm should also be studied further. Hopefully, that would allow determining under what conditions it is worthwhile to use the optimal decision rule. Thus, if improvements are not expected, then the use of non-optimal algorithms would be completely justified, given that, in practice, MAP algorithms are easier to compute.

The majority of this work lead to a publication in:

- **V. Alabau**, A. Sanchis, and F. Casacuberta. On the Optimal Decision Rule for Sequential Interactive Structured Prediction. *Pattern Recognition Letters*, 33(6):2226–2231, 2012.

Bibliography

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J. Vilar. Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1):3–28, 2009.

M. Bisani and H. Ney. Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, pp. 409–412, 2004.

H. Daumé III, J. Langford, and D. Marcu. Search-based Structured Prediction. *Machine Learning*, 75(3):297–325, 2009.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2 edition, 2001.

F. Maes, L. Denoyer, and P. Gallinari. Structured prediction with reinforcement learning. *Machine Learning*, 77(2-3):271–301, 2009.

M. Mohri. Weighted automata algorithms. *Handbook of weighted automata*, pp. 213–254, 2009.

J. Oncina. Optimum Algorithm to Minimize Human Interactions in Sequential Computer Assisted Pattern Recognition. *Pattern Recognition Letters*, 30(6):558–563, 2009.

D. Pallett, J. Fiscus, W. Fisher, J. Garofolo, B. Lund, and M. Przybocki. 1993 benchmark tests for the ARPA spoken language program. Proc. of the Workshop on Human Language Technology (HLT'94), pp. 49–74, 1994.

L. Rodríguez, F. Casacuberta, and E. Vidal. Computer Assisted Transcription of Speech. In *Proc. of the 3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'07)*, pp. 241–248, 2007.

L. Rodríguez, A. Revuelta, I. García-Varea, and E. Vidal. Interactive Text Generation for Information Retrieval. In *Proc. of the 10th International Workshop on Pattern Recognition in Information Systems (PRIS'10)*, pp. 62–71, 2010.

J. A. Sánchez, M. A. Rocha, and V. Romero. Efficient Computation of the Derivational Entropy in Word Graphs. *Submitted to Pattern Recognition Letters*, 2012.

R. Sánchez-Sáez, J. A. Sánchez, and J. M. Benedí. Interactive Predictive Parsing. In *Proc. of the 11th International Conference on Parsing Technologies (IWPT'09)*, pp. 222–225, 2009.

R. Schlüter, T. Scharrenbach, V. Steinbiss, and H. Ney. Bayes Risk Minimization using Metric Loss Functions. In *Proc. of the 9th European Conference on Speech Communication and Technology (Eurospeech'05)*, pp. 1449–1452, 2005.

R. Schlüter, M. Nussbaum-Thom, and H. Ney. On the relation of Bayes Risk, Word Error, and Word Posteriors in ASR. In *Proc. of the 11th Annual Conference of the International Speech Communication Association (Interspeech'10)*, pp. 230–233, 2010.

R. Schlüter, M. Nussbaum-Thom, and H. Ney. Does the Cost Function Matter in Bayes Decision Rule? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2): 292–301, 2012.

A. Toselli, E. Vidal, and F. Casacuberta, editors. *Multimodal Interactive Pattern Recognition and Applications*. Springer, 2011.

A. H. Toselli, V. Romero, M. Pastor, and E. Vidal. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.

E. Vidal, L. Rodríguez, F. Casacuberta, and I. García-Varea. Interactive Pattern Recognition. In *Proc. of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI'07)*, pp. 60–71, 2007.

F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 9(3):288–298, 2001.

Chapter 4

Active Interaction for Structured Prediction

Chapter Outline

4.1 Introduction

We saw in Chapter 3 that passive interaction may help to reduce the human effort in correcting the system output. Whereas this can improve the productivity of users performing their tasks, they still have to supervise the whole data set to ensure a high level of quality. That might be prohibitive in some scenarios where the company needs a small turnaround time or when the budget is limited. Fortunately, automatic systems may have information of whether their output is correct or not, and thus, direct the user towards the parts of the output where the system is not confident enough. As in this case the system takes the initiative to propose what output elements need to be corrected, we call this problem *active interaction* [Oncina and Vidal, 2011; Toselli et al., 2011], or more specifically, *active interaction for structured prediction* (AISP).

We can differentiate between two levels of AISP. First, we refer to structure level AISP when the system retrieves a full structure (object) for the user to supervise. This kind of AISP is especially valuable since it allows the user to avoid supervising the whole set of structures. Instead, the user is given hints of what structures may need supervision. Therefore, if the system is precise enough the user can save effort and, at the same time, the quality of the output improves. When working at structure level, AISP can be understood as a *quality estimation* (QE) problem. For example, in machine translation, QE is used to identify the translations that probably need to be post-edited by the professional translator [Callison-Burch et al., 2012].

On the other hand, in element level AISP, we depart from a given structure[1]. Then, the system retrieves an element (label) of the output structure and asks the user to accept it or amend it. In this case, the system can leverage structural properties of the output structure to propagate the correction to other elements of the same structure, in a similar way to the *sequential interactive structured prediction* (SISP) systems in Chapter 3.

For instance, Oncina and Vidal [2011] used an AISP technique to improve the output of the chromosome classification problem described in Section 2.2.2. The proposed strategy retrieved first to be supervised the *most confident* label, i.e., the output label with maximum posterior probability. However, they approximated the posterior probability by conditioning only on the input image that was assigned to it instead of the whole input structure. In the end, their approach required less corrections than would be necessary in the post-editing and SISP approaches. By contrast, [Culotta et al., 2006] the user was asked to amend the *least confident* element instead of the *most confident* approach by [Oncina and Vidal, 2011]. Unexpectedly, in their experiments the AISP strategy did not obtain significant improvements when compared to a random strategy. Later on in the same paper they admitted that correcting

[1] This structure can be retrieved sequentially, by using structure level AISP, or by other means, depending of which strategy is more appropriate.

low confidence elements are not likely to propagate corrections to other elements with high confidence, since they do not provide enough 'intertia' for the propagation. Conversely, correcting an element for which the system has high confidence makes the system reformulate its confidence with respect to the other elements, which facilitates correction propagation.

Other works that are also considered interactive at element level are [Serrano et al., 2010] and [González-Rubio et al., 2010]. for handwritten text recognition and machine translation, respectively, they used the *least confident* strategy to direct user's attention towards the parts of the sentence that needed corrections. These works did not leverage correction propagation, but used AISP instead to limit the supervision effort. After a batch of structures was post-edited by the user, these post-edited sentences were used to retrain the statistical models to improve the accuracy of the system. The results were quite encouraging since user effort could be reduced significantly while the transcription and translation error was kept in a reasonable level.

In this chapter we contribute to AISP in the following aspects. First, we define an optimum strategy for structure level AISP under the decision theory framework. In addition, we borrow other well-motivated strategies from active learning and compare the results against a diverse set of structured prediction tasks. Regarding element level AISP, we focus on reducing the number of corrections, instead of reducing supervision, by leveraging correction propagation. Here, we propose a set of strategies based on active learning ones and use the strategy in [Oncina and Vidal, 2011] without the approximation. Moreover, we compare these strategies with other ones from [Culotta et al., 2006]. As a result, we provide further evidence that, in order to propagate the feedback correction, we should aim to retrieve elements with high confidence first. All these strategies can be computed over word graphs. Finally, we report the element level experiments on the sequential labeling problems.

4.1.1 Active interaction is not active learning

The insightful reader would have noticed that active interaction is related to active learning [Settles, 2010] in that the system is able to propose a sample for the user to correct. However, there are some major differences. First, whereas in active learning the goal is to obtain better models, in active interaction the goal is to minimize the effort to obtain a predefined degree of error or to maximize the accuracy in a given amount of time. In other words, active learning assumes a scenario where there exist bad models and abundant unlabeled data. As the labeling of training data is expensive, active learning aims to train better models with less samples, with the final goal to minimize classification error. Conversely, active interaction assumes the existence of reasonably good but imperfect models that cannot be improved merely by using a few more data. In this case, the problem is an abundant quantity of test data, for which a high quality labeling is desired. However, it is expensive to revise and amend

the output of automatic systems so the ultimate goal of active interaction is to reduce error in the output with less effort for the human operator. In summary, the former aims at learning better models whereas the latter's goal is to reduce post-editing effort without updating the models. Note that both strategies can be compatible, i.e., we could aim at a high quality output and leverage user interactions to train better models.

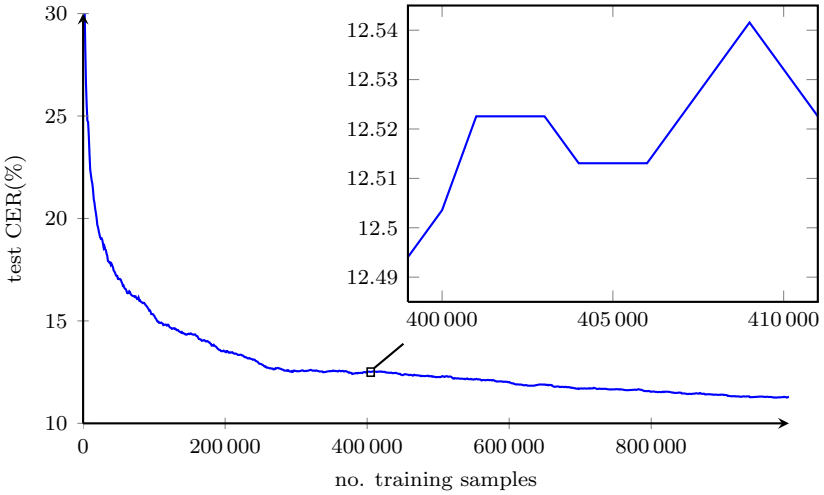


Figure 4.1: Learning curve of a handwritten digit recognition task using a k-NN classifier. The plot displays the evolution of the classification error rate for an independent test set as the number of training samples increases. Additionally, a zoomed box shows the details of the curve when adding training samples from 400000 to 410000. We can observe a non-statistically significant increase in CER.

Still, it could be argued that active learning could also achieve the goal of active interaction. As Turchi et al. [2012] noticed, the quality of the models in machine translation increases logarithmically with the amount of training data to a certain point where performance starts to decay. Simply put, the translation quality improved 'just' slightly when the training data doubled in size. The same can be observed in our DNI recognition task[2] in Figure 4.1. Each point in the curve represents classification error of an independent test for n training samples. We used a k-NN classifier trained with the digits of the DNI task. In the big picture, it can be observed that the curve rapidly stabilizes around 300000 training samples. After that point, many training samples are needed just to improve a little bit the classification error. In fact, suppose that we have a system trained with 400000 samples, and we receive an order to transcribe 10000 handwritten digits. We can see from the zoom in Figure 4.1 that error rate is practically stable, though it presents a minor, not

[2]The DNI task is an OCR problem for recognizing handwritten national identity card numbers (cf. Section 2.2.1).

statistical significant, increase of error, and not decrease as one should expect, by adding such samples to the model. In this situation, active learning cannot help building better models, an thus, there is no guarantees that by using active learning it will require less effort to reach a high quality output. In contrast, active interaction aims precisely at what we want to achieve.

Therefore, we believe that active interaction has its own purpose and deserves more attention. Nevertheless, we acknowledge that active interaction can leverage the knowledge learned in active learning to meet its goals. Hence, throughout this chapter we will analyze different strategies for AISP borrowed from active learning. Finally, we will present results on several AISP tasks.

4.2 Taxonomy of active interaction

Settles [2010] presented a survey of active learning methods, where the active learning techniques were grouped following a taxonomy. In Figure 4.2, that taxonomy has been adapted to the active interaction scenario. We can consider three aspects:

Query level: what is the object of the query. We can differentiate into *structure queries*, which aim to retrieve full objects ($\boldsymbol{y}$) or *element queries*, which aim to retrieve a single element (y_k) from a given structure ($\boldsymbol{y}$).

Scenario: the conditions in which the data can be accessed by the system, e.g., the system can inspect any structure of a set of structures or it is restrained to follow sequential order. In the *pool-based selection* scenario, all the samples are accessible at any moment, so the system should decide which samples need to be amended first. On the other hand, in the *stream-based selection* scenario the samples arrive sequentially to the system which needs to decide if the sample requires user intervention or has enough quality to pass unsupervised. Finally, in *membership query synthesis*, the samples presented to the user are synthesized. While this can be a sound technique for active learning, in active interaction it makes little sense since we want the user to label a given dataset.

Query strategy: how the system makes the decision regarding which structures or elements are to be amended by the human operator. The strategies can be motivated by statistical decision theory, information theory, by agreement of a committee of experts, or by estimated error reduction among others. Note that not all strategies are well suited for the SP problem we deal with this thesis. Thus, although they can probably be useful for other tasks, such strategies have been crossed-out in Figure 4.2 and will not be evaluated.

The following sections are devoted to study the two major groups that can be derived from the taxonomy. On the one hand, Section 4.3 delves into structure

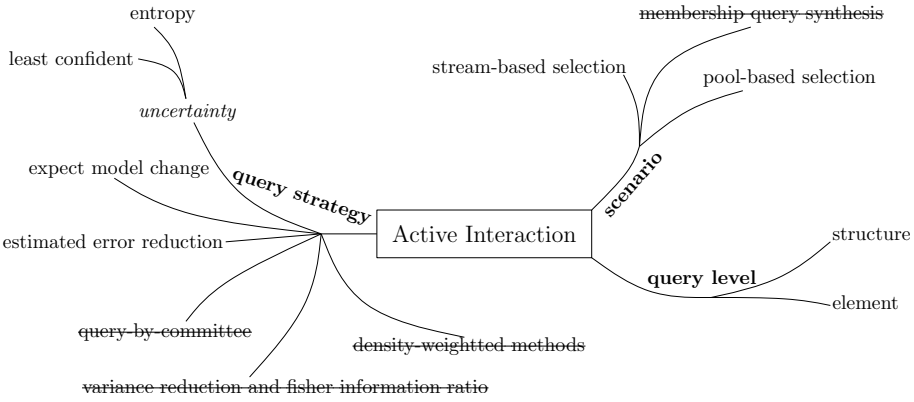


Figure 4.2: Taxonomy of active learning built upon the information in [Settles, 2010]. The techniques that have been crossed-out have not been considered in this work but they could also be used for active interaction.

level AISP, which can be either *stream-based* or *pool-based*. An optimal strategy is described and compared to other strategies. On the other hand, Section 4.4 deals with *pool-based* element level AISP. In this case, we will adapt a wider variety of strategies based on active learning. These strategies will be compared both maximizing and minimizing.

4.3 Active interaction at structure level

Often, the most sensible way to amend the labels of structured output problems is to do it at structure level, i.e., all the elements of the structure should be corrected before attempting to amend another structure. For instance, when post-editing machine translation output, one would expect the translator to pick a sentence and post-edit it completely. It would be unusual that the translator jumped from one sentence to another, post-editing only pieces of the sentences, just to come back later to finish the remaining errors. Thus, in a structured level AISP, the system would select an input and the corresponding output structure, and would ask a user to supervise it and correct it if necessary. When the user validates the corrections, the user receives another structure to amend from the system, until the budget limit B is reached. The budget limit can be understood as a maximum number of structures to supervise or as a limit for the turnaround time, among other possible restrictions. In one way or the other, the budget can be defined, in the end, as the number of structures to supervise. Henceforth, we will assume this definition for the sake of simplicity.

Let $\mathcal{D}$ be a collection of structures, such that $\mathcal{D} = \{(\boldsymbol{x}_1, \hat{\boldsymbol{y}}_1), (\boldsymbol{x}_2, \hat{\boldsymbol{y}}_2), \ldots, (\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n), \ldots, (\boldsymbol{x}_N, \hat{\boldsymbol{y}}_N)\}$, i.i.d. according to $\Pr(\boldsymbol{x}, \boldsymbol{y})$. Typically, $\hat{\boldsymbol{y}}_n$ has been

obtained by an automatic system[3] like those explained in Section 1.3 and thus may contain errors. As we would like to obtain output structures with high quality, we should use expert users to amend the outputs, which can be quite costly. Unfortunately, our budget only allows us to supervise a subset of $B \in \mathbb{N}$ structures. Thus, we say that B constitutes our budget limitation. Hence, the problem is to obtain $\mathcal{A}_B = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_b, \boldsymbol{y}_b), \ldots, (\boldsymbol{x}_B, \boldsymbol{y}_B)\}$ with $\mathcal{A}_B \subseteq \mathcal{D}$.

In AISP at structure level, we can identify two groups of structures: the group that should be amended by the user, limited by the budget, and the group that will have been processed only automatically, that is, the user will not supervise them. We will assume that the first group will have zero errors after user correction, whereas the second group may have errors produced by the automatic system. We will call the errors of the second group *residual errors*. For each group of structures we can define a loss function, λ_a that accounts for the cost of amending the first group, and λ_e that accounts for the cost of leaving the remaining structures unsupervised. Then, the final loss function can be explained by a linear combination of both loss functions, i.e., $\lambda = \lambda_a + \lambda_e$. Given that the samples in $\mathcal{D}$ are independent by definition, the decision problem can be expressed as:

$$\hat{\mathcal{A}}_B = \underset{\mathcal{A}_B \subseteq \mathcal{D}}{\arg\min} \left(\sum_{(\boldsymbol{x}, \hat{\boldsymbol{y}}) \in \mathcal{A}_B} \mathrm{R}_a(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) + \sum_{(\boldsymbol{x}, \hat{\boldsymbol{y}}) \in \mathcal{D} - \mathcal{A}_B} \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \right) \tag{4.1}$$

where R_a is the risk for the structures that will be amended and R_e the risk for the ones that will not be supervised.

Proposition 4.1. *Given a budget B, an optimal subset of structures to supervise, $\mathcal{A}_B$, can be retrieving by obtaining the bottom B samples sorted with the following criteria:*

$$\mathcal{S}(\boldsymbol{x}, \hat{\boldsymbol{y}}) = R_a(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) - R_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.2}$$

That is, we can obtain $\mathcal{A}_B$ using the following rules:

$$\hat{\mathcal{A}}_0 = \emptyset \tag{4.3}$$

$$\hat{\mathcal{A}}_{n+1} = \hat{\mathcal{A}}_n \cup \underset{(\boldsymbol{x}, \hat{\boldsymbol{y}}) \in \mathcal{D} - \hat{\mathcal{A}}_n}{\arg\min} \; R_a(\boldsymbol{y} \mid \boldsymbol{x}) - R_e(\boldsymbol{y} \mid \boldsymbol{x}) \tag{4.4}$$

Proof. The proof is by induction on B. The proposition is obviously true for $B = 0$. Then, if we assume that Equation (4.4) holds for $B = n$, we can prove

[3] $\hat{\boldsymbol{y}}_n$ could also had been generated by humans, e.g., with a crowdsourcing tool or by amateur users that need supervision.

by induction that it also holds for $B = n + 1$,

$$\hat{\mathcal{A}}_{n+1} = \operatorname*{arg\,min}_{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{A}_{n+1}} \sum_{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{A}_{n+1}} \mathrm{R}_a(\boldsymbol{y} \mid \boldsymbol{x}) + \sum_{(\boldsymbol{x}',\hat{\boldsymbol{y}}')\in\mathcal{D}-\mathcal{A}_{n+1}} \mathrm{R}_e(\hat{\boldsymbol{y}}' \mid \boldsymbol{x}') \tag{4.5}$$

$$= \operatorname*{arg\,min}_{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{A}_{n+1}} \mathrm{R}_a(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) + \sum_{(\boldsymbol{x}'',\hat{\boldsymbol{y}}'')\in\hat{\mathcal{A}}_n} \mathrm{R}_a(\hat{\boldsymbol{y}}'' \mid \boldsymbol{x}'') + \sum_{(\boldsymbol{x}',\hat{\boldsymbol{y}}')\in\mathcal{D}-\mathcal{A}_{n+1}} \mathrm{R}_e(\hat{\boldsymbol{y}}' \mid \boldsymbol{x}') \tag{4.6}$$

By definition, $\hat{\mathcal{A}}_n$ has the minimum risk for the first n structures, then we can extract it from the arg min:

$$\hat{\mathcal{A}}_{n+1} = \hat{\mathcal{A}}_n \cup \operatorname*{arg\,min}_{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n} \mathrm{R}_a(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) + \sum_{(\boldsymbol{x}',\hat{\boldsymbol{y}}')\in\mathcal{D}-\mathcal{A}_{n+1}} \mathrm{R}_e(\hat{\boldsymbol{y}}' \mid \boldsymbol{x}') \tag{4.7}$$

$$= \hat{\mathcal{A}}_n \cup \operatorname*{arg\,min}_{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n} \mathrm{R}_a(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) - \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) + \sum_{(\boldsymbol{x}',\hat{\boldsymbol{y}}')\in\mathcal{D}-\hat{\mathcal{A}}_n} \mathrm{R}_e(\hat{\boldsymbol{y}}' \mid \boldsymbol{x}') \tag{4.8}$$

As the last term of Equation (4.8) is constant for every $(\boldsymbol{x}, \boldsymbol{y})$ we reach Equation (4.3). □

A nice property of Proposition 4.1 is that we do not need to predefine a budget. Since structures can be chosen one by one, the user could decide when to stop supervising structures on the run, or she could specify time constraints and supervise as many structures as possible in such amount of time. Even in this case, Proposition 4.1 guarantees that each structure is chosen by means of an optimal decision.

Also note that this approach is only possible in the pool-based selection scenario. Since we need to sort $\mathcal{D}$, we need to have access to the whole set of structures in $\mathcal{D}$. On the other hand, the stream-based selection scenario can be regarded as a problem of trade-off between recognition error and rejection [Chow, 1970]. That problem consists in identifying which samples have errors so they should be rejected. In our case, instead of rejecting the samples they should be sent to the user for supervision, but theoretically both problems can be treated in the same way. Thus, as Chow [1970] proved, the optimal decision rule when including rejection in the classification problem is to establish a threshold over the posterior probability. In this thesis, for stream-based selection we will establish a threshold for each one of the strategies. Precisely, one of them will be the posterior probability.

4.3.1 Optimal decision in AISP at structure level

Defining the loss functions λ_a and λ_e for AISP is more involving than it was for post-editing and SISP, especially since both loss functions measure different kinds of costs. While λ_a should measure the cost of supervising and amending the output, λ_e should measure the costs incurred by leaving errors in the output. For instance, let us suppose that we have a company that produces subtitles for videolectures. For λ_a we can have the intuition that it is a matter of how much money we are willing to spend to perform the supervision. However, for λ_e, how can we measure the cost incurred by a user reading a text that has some words wrongly transcribed? In a real-life company, eventually this will be measured as an economic cost: the cost produced by the customer or consumer being convinced of the quality of the service, and thus, paying the services provided. Unluckily, to compute such sort of cost function we would need data annotated with real costs [Settles et al., 2008], which is not often the case. Anyway, here we need to make similar assumptions regarding the loss function than we made in Section 1.3. That is, if a structure is equally wrong regardless of the individual errors committed, then the *zero-one* loss function could be used. Conversely, if a structure with more individual errors is regarded as more erroneous than a structure with few individual errors, then the post-editing loss functions (Hamming or edit distance) could be used.

In principle, it seems reasonable to assume the same loss function for λ_a and λ_e, since a similar effort should be made to amend a structure than to *use* an erroneous structure. Note, however, that by following Proposition 4.1 if both rules are the same, then $\mathcal{S}(\boldsymbol{x}, \hat{\boldsymbol{y}})$ is zero for all $(\boldsymbol{x}, \hat{\boldsymbol{y}}) \in \mathcal{D}$.[4] As a result, it would be indifferent what $(\boldsymbol{x}, \hat{\boldsymbol{y}})$ is chosen. Nevertheless, unlike the passive scenario where all structures were expected to be corrected, in the active scenario we have a limited budget for the number of structures that the user can supervise. It seems reasonable to assume that just by preparing to amend a structure the user has to spend a somehow constant cognitive effort regardless of the errors committed by the system. This effort can be greater than the effort to correct an erroneous output if the corrections are easy to perform, e.g., when correcting OCR characters. However, the amending effort can dominate the final cost if the task is more cognitively demanding, as it is the case, for instance, of correcting machine translation output.

Here, we will assume that if a company has decided to spend a budget to amend some of the errors, then it values more the λ_e than λ_a, say that $\lambda_a = \alpha \cdot \lambda_e$ for $\alpha < 1$.

Proposition 4.2. *If $\lambda_a = \alpha \cdot \lambda_e$ for $\alpha < 1$, then the optimal subset of structures to supervise $\mathcal{A}_B$ is equivalent to obtain the top B samples sorted by the following*

[4] It should be pointed out that the supervision and correction of $\mathcal{A}_B$ could be performed with SISP techniques. In that case, λ_a and λ_e would not cancel out each other. Still, balancing λ_a and λ_e would be needed. For now, we will assume that even in this case, λ_a is proportional to λ_e.

criteria:

$$\mathcal{S}'(\boldsymbol{x}, \hat{\boldsymbol{y}}) = R_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.9}$$

That is, we can obtain $\mathcal{A}_B$ using the following rules:

$$\hat{\mathcal{A}}_1 = \underset{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}}{\arg\max}\, R_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.10}$$

$$\hat{\mathcal{A}}_{n+1} = \hat{\mathcal{A}}_n \cup \underset{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n}{\arg\max}\, R_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.11}$$

Proof. From Proposition 4.1 we have that

$$\hat{\mathcal{A}}_{n+1} = \hat{\mathcal{A}}_n \cup \underset{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n}{\arg\min}\, \mathrm{R}_a(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) - \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.12}$$

$$= \hat{\mathcal{A}}_n \cup \underset{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n}{\arg\min}\, \alpha \cdot \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) - \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.13}$$

$$= \hat{\mathcal{A}}_n \cup \underset{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n}{\arg\min}\, (\alpha - 1) \cdot \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.14}$$

As $\alpha < 1$, then $(\alpha - 1)$ is always a negative constant,

$$\hat{\mathcal{A}}_{n+1} = \hat{\mathcal{A}}_n \cup \underset{(\boldsymbol{x},\hat{\boldsymbol{y}})\in\mathcal{D}-\hat{\mathcal{A}}_n}{\arg\max}\, \mathrm{R}_e(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.15}$$

□

What Proposition 4.2 indicates is exactly what common sense suggests, i.e., if you want to achieve a high quality output but you have a limited budget, amend the structures that have more errors first.

4.3.2 Strategies for AISP at structure level

The optimum decision rule proposed in Proposition 4.2 is a general decision rule that must be specified for the loss function of interest for each task. Also, the active learning strategy has proposed some strategies that can be adapted to AISP. All these strategies are defined in this section and will be used in the experimentation.

Optimum strategies Under the assumption that we should use the *zero-one* loss function, Proposition 4.2 results in the *least confident* strategy [Culotta and McCallum, 2005] identified as the *uncertainty* strategy in Figure 4.2. Thus, first we build each $(\boldsymbol{x}, \hat{\boldsymbol{y}})$ pair such that

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\arg\min}\, \mathrm{R}_{01}(\boldsymbol{y} \mid \boldsymbol{x}) = \underset{\boldsymbol{y}}{\arg\max}\, \Pr(\boldsymbol{y} \mid \boldsymbol{x}) \tag{4.16}$$

and then, we find the $\boldsymbol{x}$ for which the classification is least confident following Equation (4.10),

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathcal{D}} \mathrm{R}_{01}(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \Pr(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.17}$$

Similarly, if we use a post-editing loss function, e.g., the Hamming loss,[5] we obtain the *estimated error reduction* strategy in [Settles, 2010]. First we obtain the optimum $\hat{\boldsymbol{y}}$ for each $\boldsymbol{x}$ as in Equation (1.15),

$$\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y}} \mathrm{R}_H(\boldsymbol{y}) = \arg\max_{\boldsymbol{x} \in \mathcal{D}} \sum_{i=1}^{|\boldsymbol{x}|} \Pr(y_i \mid i, \boldsymbol{x}) \tag{4.18}$$

and then we retrieve the $\boldsymbol{x}$ with the highest expected error,

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathcal{D}} \mathrm{R}_H(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) = \arg\min_{\boldsymbol{x} \in \mathcal{D}} \sum_{i=1}^{|\boldsymbol{x}|} \Pr(\hat{y}_i \mid i, \boldsymbol{x}) \tag{4.19}$$

Strategies borrowed from active learning Finally, we borrow from active learning an *uncertainty* strategy based on *entropy* [Dagan and Engelson, 1995],

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} \mathrm{H}(Y \mid \boldsymbol{x}) = \arg\max_{\boldsymbol{x}} -\sum_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{x}) \log\big(\Pr(\boldsymbol{y} \mid \boldsymbol{x})\big) \tag{4.20}$$

where $\mathrm{H}(\boldsymbol{x})$ ranges over all possible $\boldsymbol{y}$. In addition, we can use the *smallest margin* strategy [Scheffer et al., 2001],

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} \Pr(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) - \Pr(\bar{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{4.21}$$

where $\bar{\boldsymbol{y}} = \arg\min_{\boldsymbol{y} \in \{\mathcal{Y} - \hat{\boldsymbol{y}}\}} \Pr(\boldsymbol{y} \mid \boldsymbol{x})$ is the second best decoding of $\boldsymbol{x}$. In both cases, $\hat{\boldsymbol{y}}$ is obtained by minimizing the post-editing effort.

4.3.3 Experimentation

In this section, we carried out a set of experiments to evaluate how user's effort could be reduced by limiting the amount of budget for supervision. In order to do so, we evaluated how the strategies defined in Section 4.3.2 performed with all tasks presented in Chapter 2. It must be noted that, for the case of the *estimated error reduction* strategy, sequence labeling problems and post-editing problems use different loss functions. The former can be solved efficiently by minimizing the expected Hamming distance as in Equation (1.15). The latter was approximated with position specific posterior probabilities [Chelba

[5] In the case of the edit distance loss we will approximate the risk with the position specific posterior probabilities [Chelba and Acero, 2005]

and Acero, 2005] since the computational cost of minimizing the expected edit distance would be prohibitive. Thus, the *estimated error reduction* strategy for these tasks should not be considered as accurate.

All the strategies were implemented to work with *word graphs* (WG) as a common representation of the output hypothesis space (cf. Chapter 2). Word graphs are very convenient since they allow us to concentrate on the algorithms while ignoring the differences in data representation of the evaluated tasks. However, we need to take some precautions when working with WGs. First, often the scores in WGs do not represent real probabilities, and thus, they need to be normalized as we did in Equation (3.21) (Section 3.5). There we stressed the importance of the posterior scaling factor, η, which was used to adjust the peakedness of the posterior probabilities in the WGs. This can alter the results of the strategies that are built by summing up the posterior probabilities of competing hypotheses. Additionally, WGs may not include the whole search space since typically this grows exponentially with the size of the input and the size of the input and output vocabularies. Therefore, we also evaluated how the density of the word graph influences the performance of the strategies. For that, we pruned the original word graphs to contain only the hypotheses that were at much π times less likely than the most likely hypothesis. To asses how the variation of these parameters affect the strategy we computed the *area under the curve* (AUC) normalized by the original number of errors times the number of labels. The AUC is computed as the integral of the residual error as the structures are supervised. Hence, the normalized AUC is an attempt to estimate the goodness of a strategy in a single value. This way, a random strategy should give a normalized AUC around 0.5 and the goal is to obtain a value as low as possible. Regarding the pruning technique we also counted the number of paths[6] remaining in the word graph after pruning, in percentage with respect to the original size of the word graph. The parameters η and π were optimized over the development sets except for the handwritten corpus which was optimized over the test set, as the development set was not present in the official corpus partition.

4.3.4 Results

Figure 4.3 shows how the normalized AUC changes as a function of the posterior scaling factor, for each of the tasks considered. First, we can observe from the DNI tasks that the optimum posterior scale factors are quite close between the different strategies. However, this difference is relative, since the range of the scores in each tasks varies. Thus, we cannot rely on this proximity to establish the robustness of the posterior scale estimation, which should be optimized for each strategy independently. Regarding the shape of the curves, we see that the *smallest margin* strategy is not affected by the scaling factor, at least in

[6] That is, the size of the biggest n-best list that can be generated from the resulting pruned word graphs

these ranges that affect the other strategies. On the other hand, we can see that the rest of strategies present clear optimum value. The exception is the karyotype classification tasks, where the problem can be attributed to a small number of samples (100).

With respect to how strategies compare to each other, we can differentiate sequence labeling problems from post-editing problems. In the DNI task, the *estimated error reduction* strategy shows the best performance. This is the expected result since this strategy is optimum for this case. However, in the karyotype task, other approaches perform better. This can be attributed to the smaller size of the corpus and to a set of word graphs with less density. In fact, the DNI task is ideal in the sense that the word graphs contain the whole search space. Thus, the *estimated error reduction* strategy can compute the exact value. The karyotype task shows a peak around $\eta = 1.5$ where the normalized AUC for the *least confident* strategy is significantly better that the rest. On the other hand, in the post-editing tasks the *estimated error reduction* strategy lies behind, probably because an approximation was used instead of the exact algorithm. In these cases, *entropy* and *least confident* strategies are on pair, although the latter performs slightly better. Finally, we observe that the *smallest margin* strategy is not competitive in any task.

The pruning results in Figure 4.4 can give us the idea of how important is the size of the word graph for the different strategies to perform optimally. Two main effect can be observed in all plots. The first and most important one is that for the strategies to obtain the best performance, small word graphs are sufficient. In most of the cases *entropy* and *estimated error reduction* suffer pruning a bit more, since the pruning factor at which these strategies stabilize is slightly higher. The exception is, as usual, the karyotype task for the reasons already mentioned. The second effect is that *smallest margin* strategy is much more dependent to pruning. This is an unexpected behavior since here we are only considering the two most likely hypotheses, which are most surely not changed by the pruning. However, what matters is the relative difference of the probabilities after pruning which is changed by the renormalization of the probabilities.

The best results from these experiments were used to obtain the tests results in Figure 4.5. The plots show two strategies that were not present in the previous figures. First, the *oracle* strategy knows the actual number of errors in $(\boldsymbol{x}, \boldsymbol{y})$. Thus, *oracle* is the best that can be obtained. None of strategy can cross to the area with the line pattern. The other strategy is the *random* strategy, which displayed as an area corresponding to 95% of the possible random runs. It was computed as the mean $\pm$ twice the standard deviation obtained from 10 random runs. Intuitively, if a strategy is found inside this area, we could say that its performance would not be better than random.

As we can observe, all strategies are better than random. The *smallest margin* presents usually the worst results, except for the post-editing tasks where *es-*

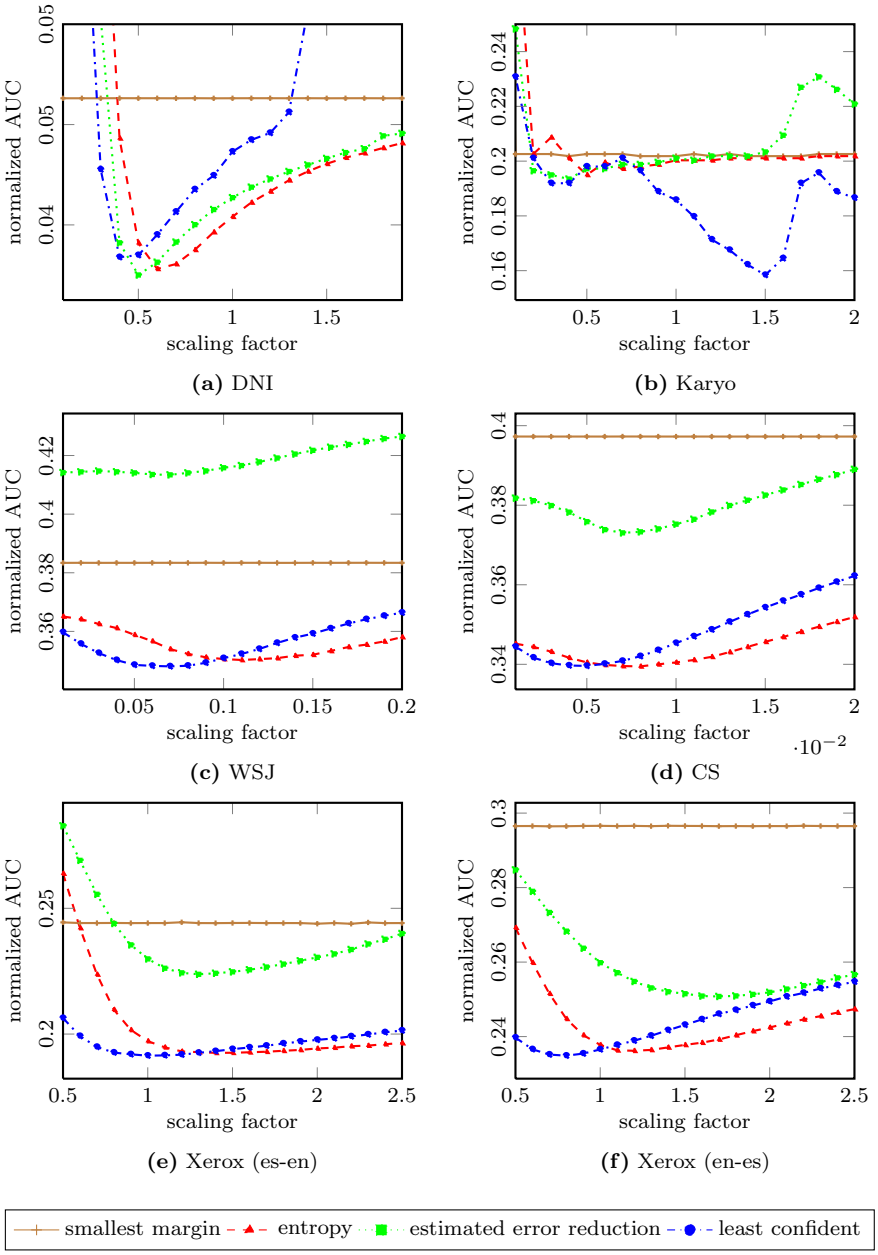


Figure 4.3: Variation of the normalized AUC as a function of the posterior scaling factor for different tasks.

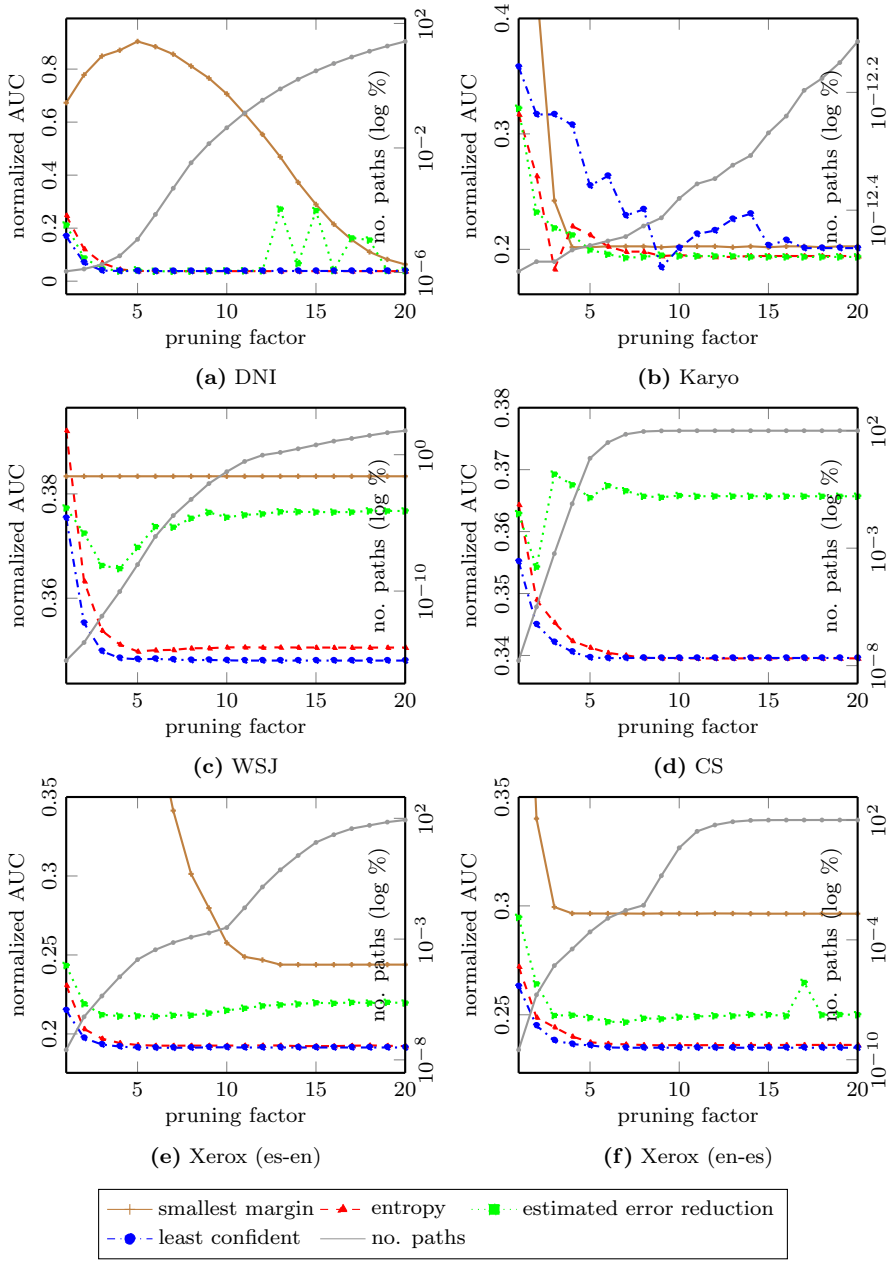


Figure 4.4: Variation of the normalized AUC as a function of the pruning threshold for different tasks. On the right axis it is indicated the percentage of paths that remain in the graph after pruning.

timated error reduction achieves worse results where just a few errors remain. With respect to the other strategies, *entropy* and *least confident* perform very similarly, being entropy slightly better. Furthermore, *estimated error reduction*, which is the optimum strategy, is the winner by little in sequence labeling tasks. Anyway, almost all approaches are closer to *oracle* than to *random* in every case, which indicate that pool-based AISP can be used effectively to reduce the supervising effort.

Finally, the results for stream-based AISP are displayed in Figure 4.6. Here the thresholds were obtained for the development set and applied in test. Hence, the points in the curve present a shift in the x-axis corresponding to the error committed when applying the threshold. The curves resemble very much that of pool-based AISP, indicating that stream-based AISP is also an interesting approach to save user effort. In particular, *entropy* seems to be consistently on pair with the best approaches for each task whereas *estimated error reduction* performs well only on sequence labeling tasks and *least confident* is usually as good as *entropy*.

4.4 Active interaction at element level

AISP at element level differs from AISP at structure level in that the system asks the user to amend single elements from a particular structure rather full structures. This fact entails significant changes in how the decisions should be made. Whereas in AISP at structure level the objects were independent among them, in AISP at element level the elements are correlated. As such, the elements follow structural properties that make correction propagation possible, as it was the case of SISP problems. Hence, this problem is well suited to attempt to minimize user corrections. Additionally, element level AISP does not assume a left-to-right scenario but the system may propose the user to correct a label at any position. Therefore, corrections might be propagated to any element, to the left or to the right, that has not been previously validated or corrected by the user. Note however that AISP is much more invasive for the user than SISP. The left-to-right assumption in SISP can be naturally accepted by users. Although users are expected to correct from left-to-right, they can actually make corrections in the prefix, e.g., the system simply disables prediction. Thus, SISP lets the user to correct the output as she wishes. On the contrary, AISP forces the user to correct a specific element in the output. Users probably would not like such imposition, particularly in those tasks that require a global view of the problem (such as translation) and thus demand an important cognitive effort to know what should be corrected. Nevertheless, there are other tasks where AISP could be accepted, such as sequence labeling tasks or handwritten text transcription, where errors are scarce or decisions can usually be taken quite locally.

A representation of an element level active system is depicted in Figure 4.7. In each iteration, the system outputs a hypothesis $\hat{\boldsymbol{y}}$ and asks the user to supervise

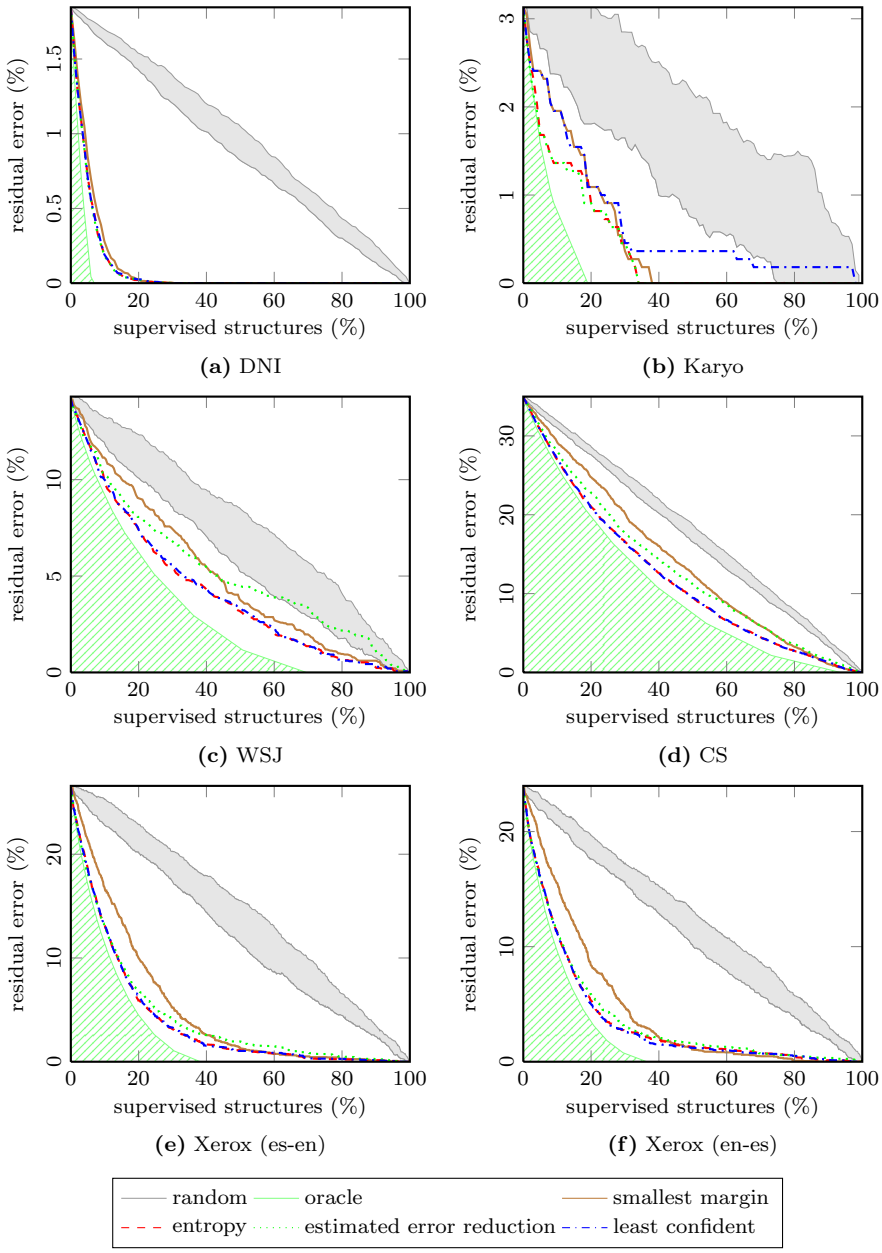


Figure 4.5: Pool-based structure level AISP results for different tasks. The grayed area represents 95% of the random strategies whereas the area with a line pattern indicates the oracle strategy and cannot be reached by any other strategy.

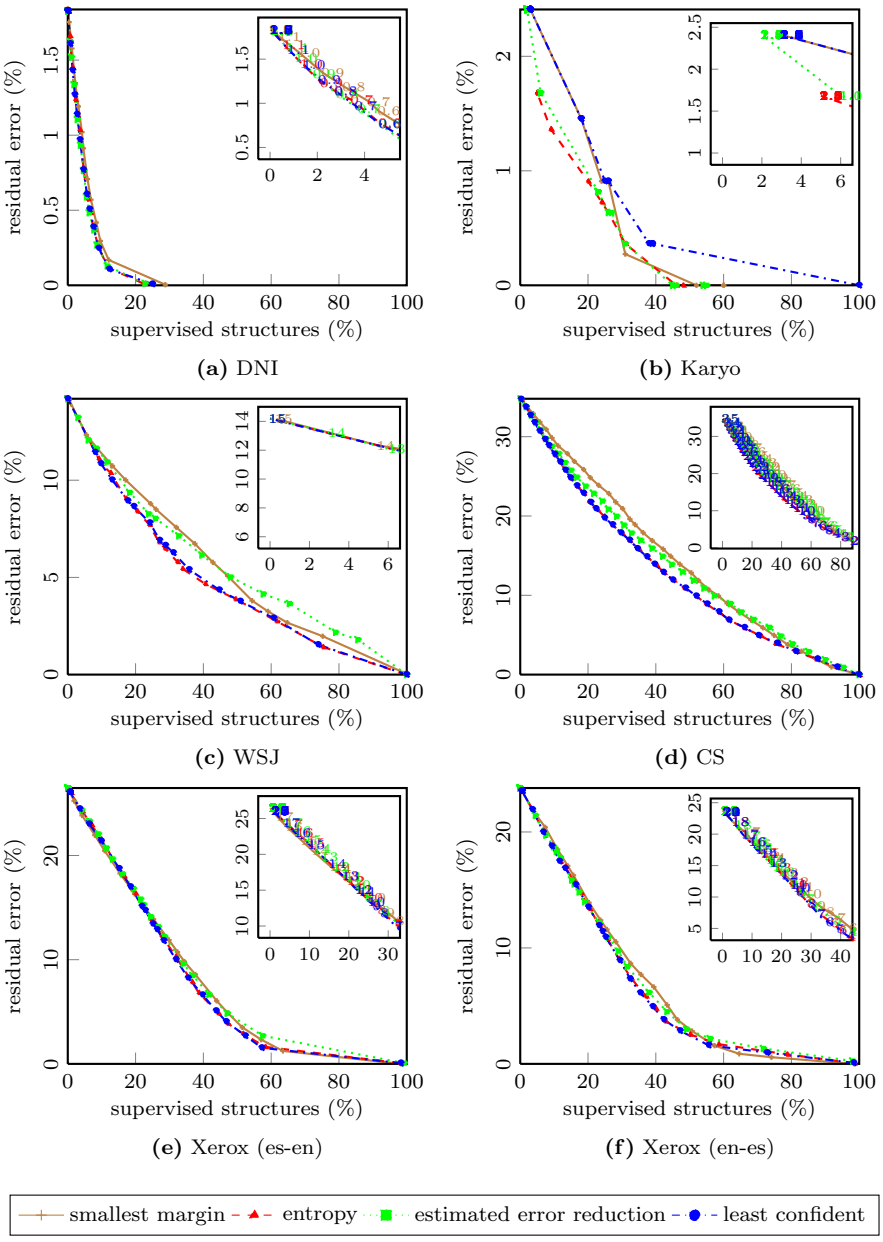


Figure 4.6: Results for stream-based structure level AISP for different tasks. The plots show the variation of the residual error as structures are supervised. The zoomed area presents the details for a set of interesting supervision thresholds.

a particular element $\hat{y}_{\hat{i}}$ of the output. Then, the user accepts the labeling if it is correct, or rejects it proposing the correct labeling with some feedback $\boldsymbol{f}$. Now, the system can propose a new hypothesis for the whole structure leveraging user's feedback. Hopefully, the new hypothesis will contain less errors as a result of the constraints imposed by the corrections made by the user. This process continues until all elements have been supervised or a given budget is reached.

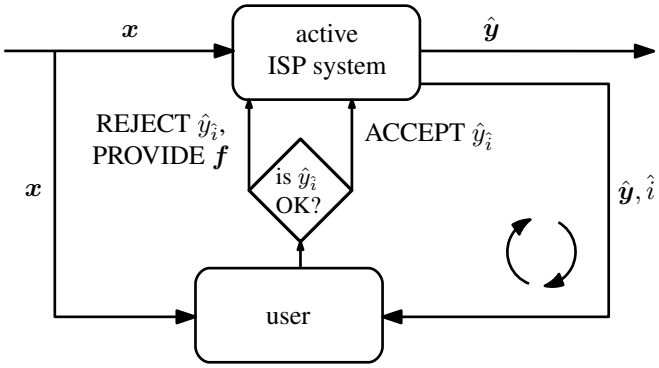


Figure 4.7: Diagram of an *active interactive structured prediction* process at element level. The system processes the input $\boldsymbol{x}$ to produce an initial output $\hat{\boldsymbol{y}}$. Then, the system selects the $\hat{i}$-th element $\hat{y}_{\hat{i}}$ for the user to analyze. The user can accept the label or reject it, in which case the correction is proposed by means of some feedback $\boldsymbol{f}$. Now, the system proposes a new hypothesis $\hat{\boldsymbol{y}}$ that is hopefully improved.

AISP at element level is closer to SISP since, as we mentioned, corrections can propagate to other elements of the structure. Therefore, in the same way than SISP, we will aim at reducing the number of corrections needed to obtain the reference. However, instead of assuming a scenario where the user reads left-to-right and fixes the fist error she encounters, in AISP the system will provide the user the label to supervise in a particular order given by a strategy $\mathcal{S}$. Anyhow, the user may have access to the whole output structure in case she needs to check it to decide which is the correct solution.

Ideally, AISP at element level could be used to retrieve a specific element for any of the structures in the collection $\mathcal{D}$, as in a global pool of elements. Nonetheless, minimizing the number of corrections can result in an increase of the supervision effort, since when accounting for corrections supervision effort is free (it has cost 0). Furthermore, it must be noted that it does not seem natural a strategy that jumps from structure to structure correcting one element at a time since that could cause confusion to the user and, probably, a loss of context of the task. Arguably, this fact can increase the cognitive effort in correcting elements and thus reduce user efficiency. Hence, we will adopt an intermediate strategy. First, AISP at structure level will be used to select the structure to amend. This way, we minimize the supervision effort at structure level. Second, AISP at element level will be used to correct the elements in the

structure. The goal now is to minimize the number of corrections to obtain the reference structure.

4.4.1 Strategies for AISP at element level

In AISP at element level we can use the same strategies that we used for AISP at structured level. Nevertheless, AISP algorithms at structure level were simpler, since the structures are independent among them. Conversely, in AISP at element level we need to check how a possible correction of the label will influence future interactions as a result of error propagation. In consequence, AISP strategies at element level should take into account all future interactions, typically by computing conditional expectations, i.e., what is the expected value of our cost function if the value of the element in position i was specified by the user. That is typically carried out by trying all the possible labels for the output element in position i. This poses a problem for tasks with large output vocabularies and for tasks where the output consists in an unknown number of elements. Hence, for simplicity, we will consider only sequence labeling problems.[7] Again, following the active learning taxonomy in Figure 4.2 we can find the strategies based on *uncertainty*, *estimated error reduction* and *expected model change*.

From the first group, the *least confident* strategy [Culotta et al., 2006] finds the element whose label was selected with the least likelihood in the structure. Thus, if we are in a sequence labeling problem and we want to minimize the labeling error we choose $\hat{y}_i$ as

$$\hat{y}_i = \arg\max_{y_i} \Pr(y_i \mid i, \boldsymbol{x}) \tag{4.22}$$

and consequently we choose $\hat{i}$ such that

$$\hat{i} = \arg\min_{i} \Pr(\hat{y}_i \mid i, \boldsymbol{x}) \tag{4.23}$$

The other strategy based on *uncertainty* is entropy. However, as we need to account for correction propagation effects that would produce a user correction, we need to retrieve the position i with maximum *conditional entropy*,

$$\hat{i} = \arg\max_{i} \mathrm{H}(Y^{-i} \mid Y_i) = \arg\max_{i} \sum_{y} \Pr(y \mid i, \boldsymbol{x})\ \mathrm{H}(Y^{-i} \mid Y_i = y) \tag{4.24}$$

where Y^{-i} represents a random variable of all the elements except the one in position i and $Y_i = y$ means that the random variable Y_i, at position i,

[7] Note that in post-editing problems the position is not a good indicator of where a element is placed since deletions and insertions may happen *before* the element as a result of correction propagation. In addition, this same fact makes it is difficult to guess whether an element is correct or not by comparing with the reference. For these reasons post-editing tasks have not been considered in this thesis for AISP at element level.

has a fixed value y. To compute $\mathrm{H}(Y^{-i} \mid Y_i = y)$ in the word graph, we can simply prune out the paths where $Y_i \neq y$ and then compute the entropy of the resulting graph.

Our experience with the *uncertainty* strategies for AISP at structure level indicates that they are competitive to reduce the supervision effort. However, as we explained, this optimization may go against reducing the number of corrections. Therefore, we will consider *certainty* strategies that are analogous to the *uncertainty* ones.

Thus, the *most confident* strategy [Oncina and Vidal, 2011] can be simply obtained by changing the arg min by an arg max,

$$\hat{i} = \arg\max_i \Pr(\hat{y}_i \mid i, \boldsymbol{x}) \tag{4.25}$$

and vice versa for *conditional harmony*, which is the opposite of *conditional entropy*,

$$\hat{i} = \arg\min_i \mathrm{H}(Y^{-i} \mid Y_i) \tag{4.26}$$

For the *estimated error reduction* technique we need to compute conditional expectations. We need to compute the Hamming risk conditioned to a given position,

$$\hat{i} = \arg\max_i \mathrm{R}_H(Y^{-i} \mid Y_i) = \arg\max_i \sum_y \Pr(y \mid i, \boldsymbol{x})\ \mathrm{R}_H(Y^{-i} \mid Y_i = y) \tag{4.27}$$

where $\mathrm{R}_H(Y^{-i} \mid Y_i = y)$ is obtained by restricting the word graph to the paths where $Y_i = y$ and then computing the Hamming risk for the resulting word graph.

Finally, [Culotta et al., 2006] introduced two strategies for corrective feedback based on *expected model change*. To begin with, *mutual information* is computed as

$$\hat{i} = \arg\max_i \mathrm{I}(Y_i; Y^{-i}) \tag{4.28}$$

They also defined the function $\#(Y_i = y)$ that returns the number of changes (as in Hamming distance) that are produced by replacing $Y_i = y$ in the most likely hypothesis. Therefore, the *expected number of changes* strategy looks for the position that, after being amended by the user, can produce more changes as a result of correction propagation,

$$\hat{i} = \arg\max_x \mathrm{E}\big(\#(Y_i)\big) = \arg\max_x \sum_y \Pr(y \mid i, \boldsymbol{x})\ \#(Y_i = y) \tag{4.29}$$

Note that here, the changes are not necessarily for good, so the strategy could find a hypothesis where already correct labels had been changed to wrong labels.

4.4.2 Experimentation

As we said, we restricted the experimentation to sequence labeling problems since the problem can be reduced to select which is the position the user has to amend. Hence, we conducted the experiments on the DNI recognition task and on the karyotype task. To do so, we first used the optimum AISP strategy at structure level for these tasks, which is the *estimated error reduction* strategy. Then, for each structure we applied the proposed element level AISP strategies. On the one hand, we measured the percentage of user corrections as the user supervises labels. On the other hand, we computed the residual Hamming error on the unsupervised labels. We used the best values for the posterior scale and pruning factor obtained in the AISP at structure level. Furthermore, for a better comparison, we also added the SISP strategy. It can be considered as a strategy that gives the elements of the structure in sequential order. For that reason it will be named *sequential*.

4.4.3 Results

The results are plotted in Figure 4.8, where the grayed out area represents the *random* strategy computed as in Figure 4.5. First, it is worth of note that, as we hypothesized, *uncertainty* strategies do not optimize user corrections since they clearly perform worse than the rest of the strategies. Especially in the case of DNI, *least confident* is far above *random*, suggesting that it is possible that optimizing supervisions is opposite to optimize corrections. Culotta et al. [2006] observed a similar behavior in their work:

> *While this may seem surprising, recall that a field will have low confidence if the posterior probability of the competing labels is close to the score for the chosen class. Hence, it only requires a small amount of extra information to boost the posterior for one of the other labels and "flip" the classification. We can imagine a contrived example containing two adjacent incorrect fields. In this case, we should correct the more confident of the two to maximize correction propagation. This is because the field with lower confidence requires a smaller amount of extra information to correct its classification, all else being equal.*

With respect to the rest of the strategies, all them are better than random for the DNI task. However, in the karyotype task the only strategy that can be found to be better that random is *most confident*. As it was the case of AISP at structure level, the karyotype task is small. Thus, it can be easier to find a good strategy by choosing random positions. Nevertheless, the *most confident* strategy still has some consistent advantage over *random*. Moreover, *expected*

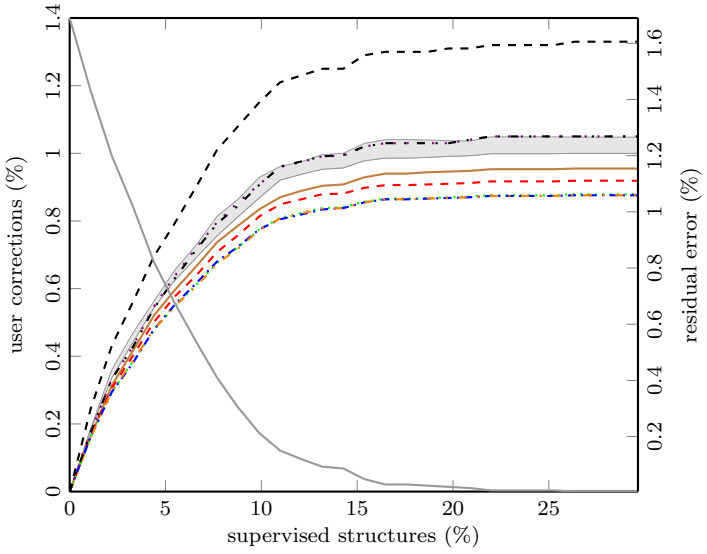
model change strategies perform alike, almost in pair with *most confident*. Additionally, in contrast with AISP at structure level, *conditional harmony* and *estimated error reduction* do not perform very well with respect to the other approaches.

Finally, the *sequential* approach can be considered as a baseline. In the DNI task, it is on pair with random, although in the upper limit. In the karyotype task, however, it is on pair with *most confident*. For what we could investigate, it appears that the way the karyotypes are labeled in a real scenario seems to be correlated, by chance, with the difficulty of the classification problem. In that case, *sequential* and *most confident* would be comparable algorithms, since they always take the most likely label. This opens an avenue to find an optimum algorithm for AISP at element level since it can be related to the *most confident* strategy.

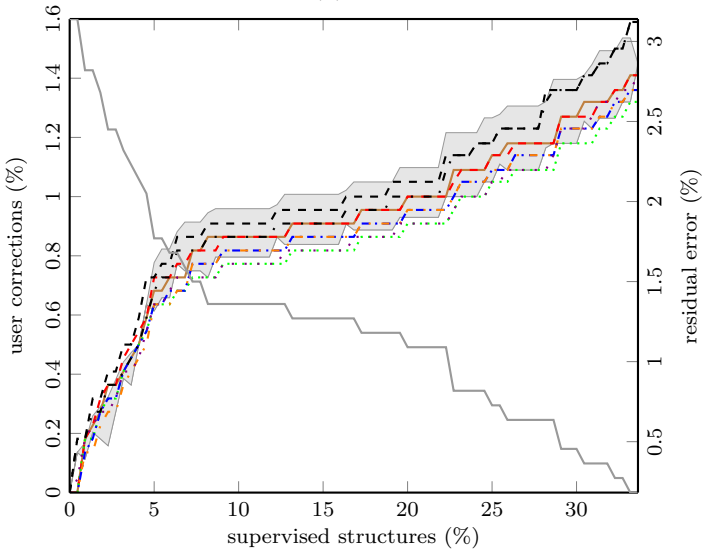
4.5 Summary of contributions

In this chapter we have studied different strategies for AISP at structure and element level. For AISP at structure level we have proved that an optimum algorithm can be build by sorting the pool of structures by the *expected error*, where the structures with highest expected error are to be amended in the first place. This strategy performed the best whenever the exact algorithm could be used. On the contrary, when an approximation was needed the *entropy* and *least confident* strategies achieved the best results. Fortunately, the experiments show that for the strategies to succeed we do not need dense word graphs. We could say that the word graphs resulting from standard decoding algorithm with standard pruning values would be just enough to compute accurate probabilities for AISP at structure level. Finally, these techniques can also be used when we do not have all the structures available, as it is the case of streaming data. Here, we showed empirically that setting a threshold over the values of the strategies is almost as effective as sorting the pool.

The second part of the chapter has been devoted to AISP at element level. In contrast to the first part, AISP at element level aimed to reduce the number of corrections instead of the number of supervisions. This was motivated by the fact that the elements in a structure present interdependencies. Therefore, correction propagation is possible. Although it may seem counter-intuitive, to reduce the number of corrections we need to find the most confident elements and not the least confident ones. The reasoning about this is that by using a *zero-one* loss function for the corrections, the supervision of a correct label is free of cost. Intuitively, when the user corrects a label which the system is very confident about it, the system needs to 'rethink' the whole hypothesis. That can give more information with regard correction propagation than a low confident label, since the system already knows that the label is probably wrong and the rest of the hypothesis should not be much affected by a change of label. Finally, we found that the *sequential* strategy performs similarly to



(a) DNI



(b) Karyo

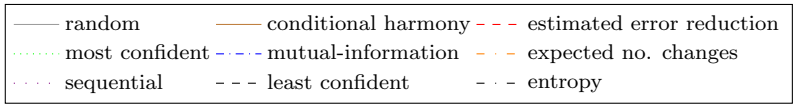


Figure 4.8: Performance different strategies to element level AISP. On the right axis it is indicated the percentage of user corrections, the lower the better. On the left axis, the residual error after having supervised and corrected the labels.

most confident when the elements in the output are sorted by the difficulty of classification. Therefore, this opens an avenue to analyze possible optimum algorithms for AISP at element level.

The work developed in this chapter is in preparation for publication.

Bibliography

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proc. of the Seventh Workshop on Statistical Machine Translation (WSMT'12)*, p. 10–51, 2012.

C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, p. 443–450, 2005.

C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

A. Culotta and A. McCallum. Reducing labeling effort for structured prediction tasks. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI'05)*, p. 746–751, 2005.

A. Culotta, T. Kristjansson, A. McCallum, and P. Viola. Corrective Feedback and Persistent Learning for Information Extraction. *Artificial Intelligence*, 170:1101–1122, 2006.

I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proc. of the 12th International Conference on Machine Learning (ICML'95)*, p. 150–157, 1995.

J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta. Balancing User Effort and Translation Error in Interactive Machine Translation Via Confidence Measures. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, p. 173–177, 2010.

J. Oncina and E. Vidal. Interactive Structured Output Prediction: Application to Chromosome Classification. In *Proc. of the 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'11)*, p. 256–264, 2011.

T. Scheffer, C. Decomain, and S. Wrobel. Active Hidden Markov Models for Information Extraction. In *Proc. of the 4th International Conference on Advances in Intelligent Data Analysis (IDA'01)*, p. 309–318, 2001.

N. Serrano, A. Sanchis, and A. Juan. Balancing Error and Supervision Effort in Interactive-Predictive Handwritten Text Recognition. In *Proc. of the 15th International Conference on Intelligent User Interfaces (IUI'10)*, p. 373–376, 2010.

B. Settles. Active learning literature survey, 2010.

B. Settles, M. Craven, and L. Friedland. Active learning with real annotation costs. In *Proc. of the NIPS Workshop on Cost-Sensitive Learning*, p. 1–10, 2008.

A. Toselli, E. Vidal, and F. Casacuberta, editors. *Multimodal Interactive Pattern Recognition and Applications*. Springer, 2011.

M. Turchi, T. De Bie, C. Goutte, and N. Cristianini. Learning to Translate: a statistical and computational analysis. *Advances in Artificial Intelligence*, p. 1–15, 2012.

Chapter 5

Online Handwritten Interaction for Machine Translation

Chapter Outline

5.1 Introduction

Since the breakout of tactile smartphones in the third quarter of 2007, the number of devices featuring a touch-screen has been increasing at a fast pace. The success of tactile smartphones has fostered a new kind of keyboardless technology which was latent until then: the tablet computers. They have been presented as a substitute of paper notebooks, as they have a similar size. However, the possibilities this new technology may provide are still to be unveiled. In that context, on-line *handwritten text recognition* (HTR) plays a crucial role. First, because to input text in such devices using a virtual keyboard is far from the efficiency of regular keyboards. Secondly, handwriting is a natural way to communicate, since it is learned early in the educational process. Withal, a HTR interface can commit recognition errors. Thus, if the HTR system is not robust enough, user experience could be negatively affected hindering its use. In this regard, many works have tried to improve HTR accuracy, primarily focusing on feature extraction and modeling [Graves et al., 2009; Jaeger et al., 2001; Liwicki and Bunke, 2006; Pastor et al., 2005].

Other authors have tackled the problem of automatically correcting errors from the system output in order to provide a more accurate input to higher-level applications. For instance, Quiniou et al. [2011] propose a technique to improve the performance of a HTR system by obtaining a consensus hypothesis out of a n-best lists, and then, characterizing the errors and correcting them. Similarly, Farooq et al. [2009] use a translation model to conduct an automatically post-editing. Finally, Shilman et al. [2006] described a user interface where handwriting and pen gestures were used as a feedback for a smart auto-completion capability. However, in this case the contextual information was not used to improve the accuracy of the HTR system. Nevertheless, those works did not leverage any contextual information of the specific task at hand, a topic that, in our opinion, has received little attention.

On the other hand, Suhm et al. [2001] proposed a multimodal dictation system that allowed the user to correct errors by respeaking, spelling or handwriting. The recognition system for the alternative modalities leveraged pre-context and post-context information from the word being corrected. Also, a bias was added towards frequently misrecognized words. Pre-context influence in accuracy was statistically significant, whereas post-context was not. The explanation for that was that post-context was frequently incorrect since users did not "select maximally contiguous regions of errors". On the other hand, the bias showed significant improvements in handwriting and spelling, but not in respeaking. Additionally, Toselli et al. [2010] explored the used of on-line HTR for interactive transcription of text images. In that work, the user was expected to correct erroneously recognized words by writing the correction using a tactile display. The authors took advantage of the erroneously predicted word and the previous one to improve HTR robustness. Latter on Pastor and

Paredes [2010] proposed the bimodal benchmark[1], where on-line and off-line HTR signals of isolated words were to be merged to improve the final accuracy. The participants could not improve the recognition accuracy aside from a late fusion by log linear combination of the on-line and off-line HTR scores.

One of the post-editing tasks that has received more attention in recent years is the correction of the output of a *machine translation* (MT) system, which has shown to boost translators productivity [Casacuberta et al., 2009; Green et al., 2013; Koehn and Haddow, 2009; Plitt and Masselot, 2010]. Typically, the correction of an MT output is performed using a keyboard and, occasionally, a mouse to position the cursor [Sanchis-Trilles et al., 2008]. Professional translators agree that this approach has been proved to be efficient. However, the user needs to be in front of a desktop computer which imposes some restrictions regarding where and how the work is to be done. Laptop computers can also be used, although arguably performance could be diminished because of the use of uncomfortable laptop keyboards and track pads. Thus, although e-pen interaction may sound impractical for texts that need a large amounts of corrections, there is a number of circumstances where e-pen interaction can be more suitable. For example, it can be well suited for amending sentences with few errors, as the revision of human post-edited sentences, or translations where the system has a high confidence that the output is of good quality. Furthermore, it would allow to perform such tasks while commuting, traveling or sitting comfortably on the couch in the living room.

In this chapter, we address the problem of using an on-line HTR system to correct the errors in an MT application, either by post-editing the translation or by interacting with the system. We propose a series of techniques that allow an early fusion of the MT and the HTR problem. The results show that important accuracy gains can be achieved due to this MT and HTR fusion. In addition, we analyze the errors committed by the system and study how a n-best list can be useful to recover from them with less user effort. Finally, we describe a series of pen gestures that can complement out HTR system, and present a preliminary research on the benefits these gestures can bring to a e-pen enabled MT interface.

5.2 Producing High-Quality Translations

In the last years, *machine translation* (MT) has become a strategic asset in the translation industry. MT is used to speed up the translation process since it enables the automatic translation of large amounts of documents. In this context, MT is approached under a statistical framework, due to the fact that statistical MT allows companies to build customized, topic-specific MT systems very economically. In MT, the problem consists in finding the most likely translation

[1] https://prhlt.iti.upv.es/page/contests/bimodal2

$\hat{\boldsymbol{y}}$ in a target language given a source sentence $\boldsymbol{x}$ in a source language,

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{x}) \tag{5.1}$$

For more information on how MT is modeled refer to Section 2.2.4.

5.2.1 Post-editing a Machine Translation Output

Although leveraging MT can be very convenient, it is usually the case that the translation quality does not meet the user requirements. Thus, the MT output must be revised. The process of revising and amending the system output, known as *post-editing* (PE), consists in deleting, inserting, substituting and swapping text from the MT output to achieve the desired quality in the translation. This is an expensive task, since the users should review the whole output and correct manually the translation errors. In the cases in which the automatically produced translations are of low quality, PE can eventually require more effort than manually translating the source input from the scratch. Moreover, in PE, the system does not take advantage of the human corrections.

5.2.2 Interactive Machine Translation

The MT paradigm is shifting slowly but steady towards an interactive MT scenario (IMT). In IMT [Barrachina et al., 2009; Foster et al., 1998; Koehn and Haddow, 2009] the system goal is not to produce translations in a completely automatic way and then perform a completely unassisted PE. On the contrary, IMT aims at building the translation collaboratively with the user as a professional advisor, so that the effort to produce a satisfactory output is minimized. In this chapter we will assume a *sequential interactive structure prediction* protocol as the one explained in Section 3.2. However, given that the results for optimum decision rule for our IMT task did not improve significantly that of the MAP decision rule (Section 3.5.1), we decided to use the later at each interaction:

$$\hat{\boldsymbol{y}}_s = \arg\max_{\boldsymbol{y}_s} \Pr(\boldsymbol{y}_s | \boldsymbol{x}, \boldsymbol{y}_p) \tag{5.2}$$

where the number of iteration $^{(i)}$ has been omitted for simplicity. Nevertheless, it should be noted that the approaches that will be explained through out this chapter can be applied to the optimum decision rule with minor changes.

5.3 Using On-Line HTR to Correct MT Output

Let us imagine an application devised to translate documents. On the one hand, there is a text area with the output of an automatic machine translation

system. As this output may contain errors, the user of the application reads the output to locate the first error. The reading is performed in a specific order, left-to-right in most western languages, for instance. Let us also assume that when the user finds the first error, all the words before it have already been revised and validated. Thus, they can be regarded as correct. Once the error has been located, the user introduces the correction with a stylus. As a result, the system receives a position where the error is located, a word that is incorrect (the word pointed by the position) and a sequence of pen strokes that represent the correct word in that position. On the other hand, the source document to be transcribed is shown to the user. There is a strong relationship among the words in the source sentence and the words in the target sentence.

Figure 5.1 is a mock-up of a possible application on a tablet device for such scenario. The screen is divided in two sections. First, the upper part shows the source document, and probably the source sentence being currently translated, $\boldsymbol{x}$, is highlighted appropriately. Second, the lower section contains the current state of the translation, $\boldsymbol{y}$. Since we assume that post-editing is usually performed from left to right, the text which has already been revised and validated, $\boldsymbol{y}_p$, is shown within a dotted box. On the other hand, the text which is to be revised, e, is displayed grayed out. From the sentence currently being translated we can identify three parts: the revised prefix of the sentence, $\boldsymbol{y}_p$, the error committed by the system, e, and the correction proposed by the user introducing strokes with a stylus, $\boldsymbol{f}$.

In a scenario as described above, the HTR subsystem should make few errors to make the application usable. The aim here is to devise a robust HTR system that allows a potential user to revise and correct the output of a machine translation system using an electronic pen. To this regard, we assume that the user will introduce the corrections by writing over the word or sequences of words (phrases) she judges to be incorrect. Thus, the problem of on-line HTR consists in converting a sequence of strokes, $\boldsymbol{f}$, into a word or phrase in text format, $\boldsymbol{d}$. The strokes can be acquired from a stylus, electronic pen or a touch-screen.

The baseline approach to the problem from a statistical point of view is to obtain the most likely decoding $\boldsymbol{d}$ given the strokes $\boldsymbol{f}$,

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}) = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d}) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{5.3}$$

where $\Pr(\boldsymbol{d})$ can be represented by a language model and $\Pr(\boldsymbol{f} \mid \boldsymbol{d})$ by morphological models. For more details on how this is modeled refer to Section 2.3.1.

Nevertheless, our purpose is to take advantage of the information available in the MT application to make on-line HTR more robust. In the remainder of this section, we will introduce gradually the different methods to make the on-line HTR system more robust.

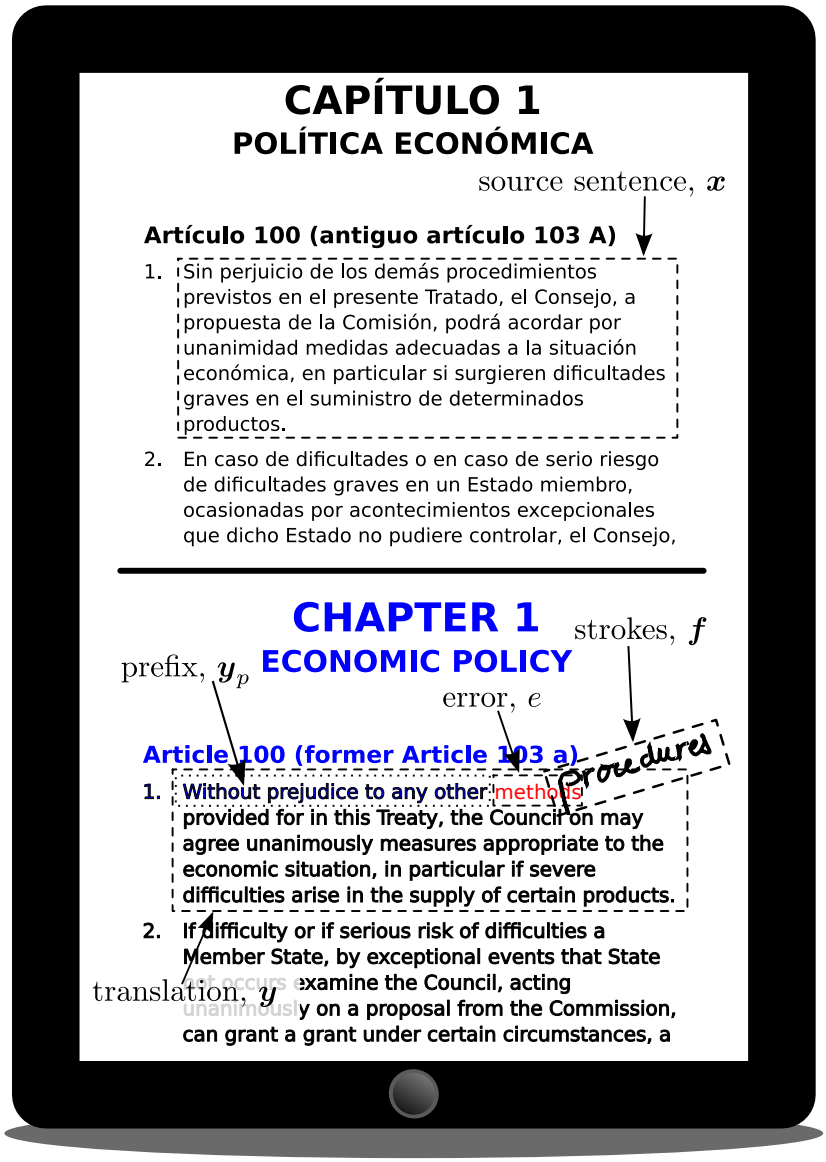


Figure 5.1: Mock-up of an interactive machine translation application on a tablet device.

5.3.1 Discarding the produced error

In the e-pen enabled interface aforementioned, the user is expected to write the strokes over the erroneously translated word, and thus, the system knowns what word the user wants to replace. Therefore, the first and easiest approach

is to remove the erroneous word e from the list of candidate hypotheses. This way, Equation (5.3) becomes

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d} \neq e} \Pr(\boldsymbol{d}) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{5.4}$$

5.3.2 Exploiting information from the revised translation

The second sensible approach to take is to add information regarding the revised translation prefix, $\boldsymbol{y}_p$. Again, from Equation (5.3) we can derive an HTR system that takes into account previously validated words:

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}, \boldsymbol{y}_p) \tag{5.5}$$

which, under the assumption that $\Pr(\boldsymbol{f} \mid \boldsymbol{d}, \boldsymbol{y}_p)$ does not depend on $\boldsymbol{y}_p$ if $\boldsymbol{d}$ is known, can be computed as

$$\hat{\boldsymbol{d}} \approx \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{y}_p) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{5.6}$$

Here, $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p)$ is a prefix language model, i.e., the probability of $\boldsymbol{d}$ depends on the left-context. Of course, we can also discard the erroneous word from Equation (5.6),

$$\hat{\boldsymbol{d}} \approx \arg\max_{\boldsymbol{d} \neq e} \Pr(\boldsymbol{d} \mid \boldsymbol{y}_p) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{5.7}$$

These techniques can be extrapolated to most post-editing tasks. In fact, Toselli et al. [2010] was the first to propose the use of the erroneous word and a 2-gram model to improve the HTR performance for interactive transcription of text images. In the subsequent sections, we present our contributions to how the information regarding the translation process can be exploited for further improved HTR decoding.

5.4 Leveraging information from the source sentence

A specific source of information that can help to improve robustness in the MT scenario is, naturally, the sentence in the source language. Since the target sentence conveys the meaning of the source sentence, $\boldsymbol{x}$, user corrections should be restricted somehow to the possible translations of it. Hence, we can formulate the problem as,

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}, \boldsymbol{y}_p, \boldsymbol{x}) \tag{5.8}$$

Then, assuming that $\Pr(\boldsymbol{f} \mid \boldsymbol{d}, \boldsymbol{y}_p, \boldsymbol{x})$ does not depend on $\boldsymbol{y}_p$ and $\boldsymbol{x}$ if $\boldsymbol{d}$ is known, Equation (5.8) can be rewritten as

$$\hat{\boldsymbol{d}} \approx \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x}) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{5.9}$$

Nevertheless, the relationship between the target and the source sentence in $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$ is not trivial to establish. We have considered two possibilities to approximate $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$. First, word-based models are the basis for modern statistical MT [Brown et al., 1993]. Although they cannot provide a good performance when translating complete sentences, they offer a smoothed and reliable probability distribution for word models. In addition, they serve as initialization for the second kind of models considered: phrase-based models [Koehn, 2010]. These models improve word-based models since they are able to translate sequences of words (phrases) and constitute the state-of-the-art in MT.

5.4.1 Word-based translation models

Brown et al. [1993] approached the problem of MT in Equation (5.1) from a statistical point of view as a search problem of a translation $\boldsymbol{y}$ (cf. Section 2.2.4). In this approach a hidden variable a is introduced that represents the alignment between the words in the source and target sentence. In this way, an alignment a is defined as a vector of length $|\boldsymbol{y}|$, in which the i-th element corresponds to the source position j, i.e., the word x_j, to whom y_i is aligned. Formally, we can model the posterior probability of the target sentence $\boldsymbol{y}$ being a translation of the source sentence $\boldsymbol{x}$ by marginalizing over the set of all possible alignments between the words in $\boldsymbol{y}$ and the words in $\boldsymbol{x}$,

$$\Pr(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{a} \Pr(\boldsymbol{y}, \boldsymbol{a} \mid \boldsymbol{x}) \tag{5.10}$$

Then, $\Pr(\boldsymbol{y}, \boldsymbol{a} \mid \boldsymbol{x})$ can be decomposed using the chain rule. After making some strong assumptions, two distributions are obtained. First, the alignment model, $Pr\,(j \mid i, |\boldsymbol{x}|)$, represents the probability of the target word at position i to be aligned with the source word at position j for a source sentence of length $|\boldsymbol{x}|$. Second, the word translation model, $\Pr(y_i \mid x_j)$, models the probability of the target word y_i to be a translation of the source word x_j. The above assumptions are necessary to make model estimation tractable and result in the so-called *model 2* (M2) [Brown et al., 1993].

In M2, the alignment probability, $Pr\,(j \mid i, |\boldsymbol{x}|)$, can be approximated by the relative frequency of position j in the source sentence to be aligned with position i in the target sentence for a source sentence of length $|\boldsymbol{x}|$. On the other hand, the translation probability, $\Pr(y_i \mid x_j)$, can be approximated by a word-to-word statistical dictionary which essentially is the relative frequency of y_i being

aligned with x_j. Nonetheless, these frequencies cannot be estimated directly since the real alignments are unknown. Thus, the EM algorithm is needed to reliably estimate these probabilities [Brown et al., 1993]. *Model 1* (M1) is a particular case of word-based models where the alignment probability is approximated by an uniform probability distribution, $Pr\,(j \mid i, |\boldsymbol{x}|) \approx (|\boldsymbol{x}| + 1)^{-1}$.

Returning to our original problem, we can approach $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$ in Equation (5.9) with word-based translation models with some assumptions. First, the direct use of M2 models would require to modify the search algorithm of the HTR decoder, probably increasing its computational complexity. For that reason, in the case of M2 we will only allow the introduction of one handwritten word[2], and thus, $|\boldsymbol{d}| = 1$ and $\boldsymbol{d} = (d_1)$. Second, from the prefix $\boldsymbol{y}_p$ we can obtain the position of the erroneous word to be corrected (obviously, also the position of d_1), $i = |\boldsymbol{y}_p| + 1$, ignoring the rest of the words in the prefix. Taking into account both considerations the first term of Equation (5.9) can be rewritten as,

$$\Pr(d_1 \mid \boldsymbol{y}_p, \boldsymbol{x}) \approx p(d_1 \mid i, \boldsymbol{x}) \tag{5.11}$$

Then, we can introduce the alignment between d_1 and the words from the source sentence by summing for every possible position j in $\boldsymbol{x}$,

$$\begin{aligned} p(d_1 \mid i, \boldsymbol{x}) &= \sum_{j=1}^{|\boldsymbol{x}|} p(d_1, j \mid i, \boldsymbol{x}) \\ &= \sum_{j=1}^{|\boldsymbol{x}|} p(j \mid i, \boldsymbol{x}) p(d_1 \mid j, i, \boldsymbol{x}) \end{aligned} \tag{5.12}$$

Finally, if we assume, in a similar way to M2, that $p(j \mid i, \boldsymbol{x})$ does not depend on $\boldsymbol{x}$ but on $|\boldsymbol{x}|$, and that $p(d_1 \mid j, i, \boldsymbol{x})$ does not depend on the whole $\boldsymbol{x}$ but just on the source word x_j aligned to d_1, then we can approximate Equation (5.12) as

$$p(d_1 \mid i, \boldsymbol{x}) \approx \sum_{j=1}^{|\boldsymbol{x}|} p(j \mid i, |\boldsymbol{x}|) p(d_1 \mid x_j) \tag{5.13}$$

where $p(j \mid i, |\boldsymbol{x}|)$ is an M1 or M2 alignment model and $p(d_1 \mid x_j)$ is a statistical dictionary.

Figure 5.2 reflects the role of the alignments and the dictionary. The source sentence is shown in the middle, and each word has its corresponding position, j, as a subscript. Above each word, there is a list of its most probable translations using the dictionary. Grey levels are proportional to the probability of the dictionary. On the other hand, in the bottom, there is a possible translation,

[2]This assumption is not necessary for M1, but we will keep it for simplicity.

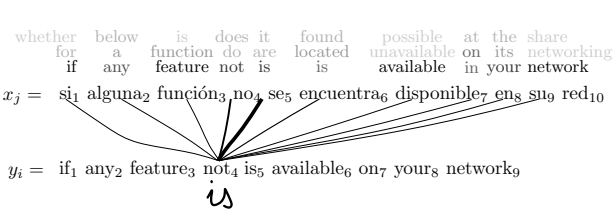


Figure 5.2: Visualization of alignments and translation dictionary.

which has an error in position $i = 4$. Below that, the user is trying to correct that mistake by introducing the word ﻳﺲ. Each link between a source word and the target word in position 4 represents the alignment probability. The stroke boldness is proportional to the M2 alignment probability. Note that for an M1 model, all alignments would have had the same thickness.

If we focus on the possible candidate transcriptions of the handwritten word ﻳﺲ, we realize that there are two possibilities that could create confusion to the decoder: 'if' and 'in' due to the fact that the strokes for 'is', 'if' and for 'in' can be very similar. Both can compete with the correct transcription 'is'. The word 'if', has a high probability in the dictionary, $p(\text{if} \mid \text{si}_1) = 0.88$, whereas other candidates have lower probabilities. Then, since the M1 model has a uniform alignment probability, it would assign a higher probability to 'if' than to 'is'. However, 'si$_1$' actually has a lower probability of being aligned with 'not$_4$'. Therefore, the M2 model is able to solve this shortcoming thanks to the alignments with high probability to the correct words. In this case, $p(5 \mid 4, 10) = 0.38$ and $p(6 \mid 4, 10) = 0.12$, whereas $p(1 \mid 4, 10) = 0.04$.

It must be noted that word dictionaries are not symmetric, i.e., $p(d_1 \mid x_j)$ is probably different to $p(x_j \mid d_1)$. As the inverse statistical dictionary can be obtained as a by-product of the standard training procedure of MT systems, we have decided to leverage the knowledge from the inverse statistical dictionary as an approximation to $p(d_1 \mid x_j)$. By applying the Bayes' rule,

$$p(d_1 \mid x_j) = \frac{p(d_1)\, p(x_j|d_1)}{\sum_{d'} p(d')\, p(x_j|d')} \tag{5.14}$$

Arguably, this model can provide a smoother probability, and, as we will see in the experiments, in practice they can provide a better vocabulary coverage. Summarizing, we propose four word-based translation models: direct M1 and M2 models, both having a direct dictionary, and inverse M1 and M2 models with the inverse dictionary from Equation (5.14).

5.4.2 Phrase-based translation models

Word-based translations provided a basis for MT. However, their performance regarding translation quality is not sufficient. Their limitation resides in that

they cannot model properly context information [Zens et al., 2002]. Phrase-based models aim at reducing this problem by translating phrases (sequences of words) instead of single words. These models were popularized by Och and Ney [2002], who established the state-of-the-art phrase-based log-linear models. Phrase-based models offer a great opportunity to estimate $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$. However, we cannot use these models directly, as occurs with word-based models. One limitation of phrase-based models is that their probabilities are 'peaky' and, usually, they cannot model all possible translations. As a result, it would be possible that $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$ takes a value 0 for a user established prefix. Then, it is necessary to smooth theses probabilities. For instance, we can generate n-gram-like models from the hypotheses in a word graph (WG) of a MT system [Ueffing et al., 2002] (cf. Section 2.1).

As has been described in Section 2.1, *word graphs* (WG) contain a set of the most likely translations of the source sentence. Although one may think that WGs could be directly used to estimate $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$, there are some details that must be taken into account. First, WGs do not contain all the possible translations since, in practice, many pruning techniques must be used to generate the translations efficiently. Second, phrase-based models are not good dealing with long distance alignments due to the introduction of heuristic length constrains and, thus, WGs do not present sentences with long distance reorderings. In those cases, a user validating a prefix $\boldsymbol{y}_p$ that is not contained in the WG would obtain a zero probability in $\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$. Hence, it is interesting to smooth the probability distribution encoded in the WGs. To do so, WGs can be simplified in the way that language modeling is typically approached: we make each word to depend only on the preceding $n-1$ words instead of depending on the whole prefix. As a result, we can rewrite the constrained language model as

$$\Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x}) \approx p(\boldsymbol{d} \mid p_{i-n+1}^{i-1}, \boldsymbol{x}) \tag{5.15}$$

where p_{i-n+1}^{i-1} are the words in the prefix from position $i-n+1$ to position $i-1$. That is, $p(\boldsymbol{d} \mid p_{i-n+1}^{i-1}, \boldsymbol{x})$ only takes into account the latest $n-1$ words from the prefix. Note that $p(\boldsymbol{d} \mid p_{i-n+1}^{i-1}, \boldsymbol{x})$ is very similar to a n-gram language model except for the dependency on $\boldsymbol{x}$. Khadivi and Ney [2008] presented a closely related approach for ASR as input to MT. In that work, n-gram-like models were generated from n-bests lists instead of WGs. The advantage of the n-gram-like prefix modeling assumption is that the models only take into account a limited size of the history, and thus, can provide a smoother probability distribution.

What follows is a procedure proposed by Campbell and Richardson [2008] to generate such n-gram-like models from the sentences in the WG. First, the posterior probabilities for each edge, $p(e|\boldsymbol{x})$, must be computed as in Section 3.4. The posterior probability for a node u, $p(u|\boldsymbol{x})$, accounts for the probability mass of the hypotheses that pass through u. This probability can be computed

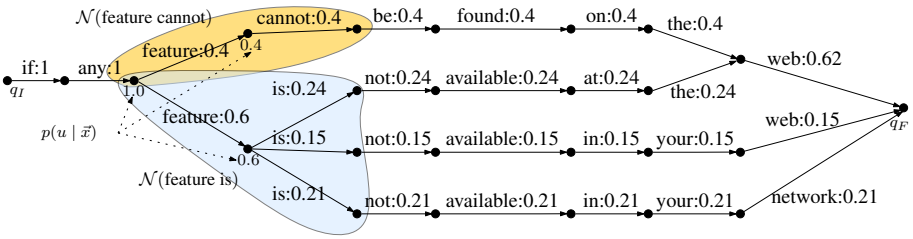


Figure 5.3: Word graph with posterior probabilities. It represents a subset of hypotheses of the hypothesis space of a state-of-the-art translation model for the source sentence 'si alguna función no se encuentra disponible en su red'. On the left, the set of links considered when computing the average count of the bi-gram 'feature is' (below) whereas the link considered for the bi-gram 'feature cannot' (above).

in a similar way to $p(e|\boldsymbol{x})$ by using the forward $\Phi(u)$ and backward $\Psi(u)$ scores,

$$p(u|\boldsymbol{x}) = \frac{\Phi(u)\Psi(u)}{\Phi(q_f)} \tag{5.16}$$

Then, the average counts of word sequences can be estimated as follows. Let $\mathcal{N}(d^i_{i-n+1})$ be a cluster of all the sequences of concatenated edges $\boldsymbol{e} = \{e_1, e_2, \ldots, e_n\}$ in the WG that generate the output d^i_{i-n+1}. That is, given that $e_j = (y_j, u_j, v_j)\ \forall j : 1 \leq j \leq n$, $\boldsymbol{e}$ meets the requirement that $y_1 y_2 \ldots y_n = d^i_{i-n+1}$. Then, for a given n-gram length, the expected count of occurrence of d^i_{i-n+1}, $C^*(d^i_{i-n+1} \mid \boldsymbol{x})$, can be computed as

$$C^*(d^i_{i-n+1} \mid \boldsymbol{x}) = \sum_{\boldsymbol{e} \in \mathcal{N}(d^i_{i-n+1})} \frac{\prod_{j=1}^{n} p(y_j|\boldsymbol{x})}{\prod_{j=2}^{n} p(u_j|\boldsymbol{x})} \tag{5.17}$$

An example of such $\mathcal{N}(\cdot)$ clusters is shown in Figure 5.3 for the 2-grams 'feature cannot' and 'feature is'. Then, $C^*(\text{feature cannot} \mid \boldsymbol{x})$ and $C^*(\text{feature is} \mid \boldsymbol{x})$ can be computed as

$$C^*(\text{feature cannot} \mid \boldsymbol{x}) = \frac{0.4 \cdot 0.4}{0.4} = 0.4$$

$$C^*(\text{feature is} \mid \boldsymbol{x}) = \frac{0.6 \cdot 0.24}{0.6} + \frac{0.6 \cdot 0.15}{0.6} + \frac{0.6 \cdot 0.21}{0.6} = 0.6$$

That is, 'feature is' appears 0.6 times in average in the possible set of translation, whereas 'feature cannot' only appears 0.4 times. Note that if a sequence of words appears more than once in a sentence, the average counts might exceed 1.

Now, n-gram-like probabilities from the word posterior WG can be calculated after a proper normalization:

$$p(d_i|d_{i-n+1}^{i-1}, \boldsymbol{x}) = \frac{C^*(d_{i-n+1}^{i} \mid \boldsymbol{x})}{C^*(d_{i-n+1}^{i-1} \mid \boldsymbol{x})} \tag{5.18}$$

Then, Equation (5.18) can be used directly in Equation (5.9) to approximate $\Pr(\boldsymbol{d}|\boldsymbol{y}_p, \boldsymbol{x})$. In other words, given a sequence of words d_{i-n+1}^{i}, $p(d_i \mid d_{i-n+1}^{i-1}, \boldsymbol{x})$ can be estimated by summing up the posterior probabilities of all sentences containing the sequence d_{i-n+1}^{i}.

Following the example in Figure 5.3 and being $C^*(\text{feature}|\boldsymbol{x}) = 1$,

$$p(\text{cannot}|\text{feature}, \boldsymbol{x}) = \frac{0.4}{1} = 0.4$$
$$p(\text{is}|\text{feature}, \boldsymbol{x}) = \frac{0.6}{1} = 0.6$$

The estimation in Equation (5.15) presents the problem that usually many n-grams are not represented in the WG. Then, they will have zero probability, and the HTR system will fail to recognize them. A common approach is to rely on simpler models to account for unseen events using back-off models [Katz, 1987]. As the estimated counts are not real counts (they vary from 0 to the number of times the n-gram occurs in a sentence), typical discount methods cannot be applied. However, absolute discount can be used [Ney et al., 1995], which consists in subtracting a constant, ϵ, from C^*. Although this is a simple discounting method, it has an interesting interpretation. Word posterior probabilities have been extensively used for computing word-based confidence measures [Sanchis et al., 2012; Wessel et al., 2001]. They usually define a threshold over the word posterior probability to identify correctly recognized words. In the same sense, ϵ establishes a threshold over the expected counts. If a count C^* is bellow ϵ, then it is considered to have low confidence and discounted to zero.

Furthermore, only words present in the WG are included into the model (which implies a high number of out-of-vocabulary words, since WGs only contain the words of the most likely hypotheses). The out-of-vocabulary (OOV) problem is solved by distributing the discounted mass from the unigram among the remaining words of the vocabulary.

Finally, to improve the estimation of unseen events, n-grams from the WG can be interpolated linearly with the standard n-gram model:

$$p_\gamma(\boldsymbol{d}|\boldsymbol{y}_p, \boldsymbol{x}) = \gamma\, p(\boldsymbol{d}|\boldsymbol{y}_p, \boldsymbol{x}) + (1-\gamma)\, p(\boldsymbol{d}|\boldsymbol{y}_p) \tag{5.19}$$

This way, the words that were not used by the MT engine are assigned a meaningful probability.

5.5 Integrated HTR and IMT decoding

Previous models assume a two-step process, in which the strokes are first decoded into a word or phrase, and then, the decoded word is used to correct the output of the IMT system. However, this decoding can be performed in an integrated way by marginalizing over every possible decoding $\boldsymbol{d}$ in Equation (5.2):

$$\hat{\boldsymbol{y}}_s = \arg\max_{\boldsymbol{y}_s} \sum_{\boldsymbol{d}} \Pr(\boldsymbol{y}, \boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{f}, \boldsymbol{x}) \tag{5.20}$$

Then, we can decompose Equation (5.20) using the chain rule. Approximating the sum by the maximum, and assuming that $\Pr(\boldsymbol{y}_s \mid \boldsymbol{y}_p, \boldsymbol{f}, \boldsymbol{d}, \boldsymbol{x})$ does not depend on $\boldsymbol{f}$ if $\boldsymbol{d}$ is known,

$$(\hat{\boldsymbol{y}}_s, \hat{\boldsymbol{d}}) \approx \arg\max_{\boldsymbol{y}_s, \boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{f}, \boldsymbol{x}) \Pr(\boldsymbol{y}_s \mid \boldsymbol{y}_p, \boldsymbol{d}, \boldsymbol{x}) \tag{5.21}$$

Note that, though not necessary, we have added $\boldsymbol{d}$ in the arg max since we are also interested in the result of the online HTR decoding.

The first term in Equation (5.21) can be approximated as in Equation (5.3), Equation (5.4), Equation (5.6), Equation (5.13) or Equation (5.15). The second term is a prefix conditioned translation model as in Equation (5.2). This probability forces $\boldsymbol{d}$ not just to be a good translation of $\boldsymbol{x}$ but to form part of a sentence that is good translation of it. Hence, the decoding of $\boldsymbol{d}$ is benefiting from a new source of information.

5.6 Experiments

In this section, we present a set of experiments to assess the performance of the HTR systems described in previous sections. Two kinds of experiments were conducted. First, the word-based experiments assume that the user only writes one word at a time. Second, in the phrase-based experiments the user writes a set of consecutive erroneous words. Additionally, two corpora were generated from the Xerox corpus, one with Spanish phrases from translations of English sentences and the other one with English phrases from translations of Spanish sentences. The details of how the two corpora were generated are given in Section 2.3.1.

In order to make the references easier, we will name the different systems as follows:

HTR. The baseline HTR system as defined in Equation (5.3).

ERR. The baseline HTR system after removing the erroneous word, Equation (5.4).

*n***PREF.** In Equation (5.6), the latest n words of validated prefix in the target sentence are taken into account.

M1. In Equation (5.13), information regarding the dictionary is used, but the alignment probabilities are uniform.

M2. In Equation (5.13), the dictionary and the alignment probabilities are used.

M1-INV. Like **M1** but with the inverse dictionary in Equation (5.14).

M2-INV. Like **M2** but with the inverse dictionary in Equation (5.14).

*n***WG.** In Equation (5.15), the system uses an n-gram that has been extracted from the phrase-based translation WG.

Furthermore, if Equation (5.21) is used, it will be marked with **+IMT**. In addition, several of the proposed systems can be combined by linear interpolation. For this case, we will add them up with the + symbol. For instance, **ERR+3PREF+M1+IMT** is a linear interpolation between a prefix 3-gram and a M1 model for which the erroneous word has been removed and IMT has been activated. Besides, in the case that a log-linear combination[Berger et al., 1996; Och and Ney, 2002; Papineni et al., 1998] is used instead of a linear interpolation, it will be remarked as so.

In addition, the proposed language models were encoded as n-grams. The aim of this is two-folded. First, we would like to leverage current HTR systems without custom software modifications. Second, since the new sources of information are added early in the HTR system, we expect to reduce the error cascade produced in post-processing error correcting systems. However, although all the proposed models can be trivially encoded as 1-grams for the case of word-based recognition, some of them cannot be encoded efficiently for n-grams as such and require special search algorithms. As these cases are out of the scope of the current work, such models will not be evaluated for phrase recognition. Nevertheless, these models could also be applied in a post-processing rescoring stage. For instance, both **M1** and **M2** models can be easily encoded as a 1-gram for word-based recognition. As there is just one possible value for i and $\boldsymbol{x}$, the 1-gram can be build by computing Equation (5.13) for each word of the vocabulary. In contrast, **M2** models cannot be encoded as n-grams for phrase recognition since the probability depends on the position i of the hypothesized word, and then, i should be stored in the search algorithm for every word hypothesis. Luckily, **M1** models assume independence of the position i so they can be encoded as a 1-gram even for the case of phrase recognition. In a similar way, the integration of HTR and IMT cannot be easily incorporated into a n-gram language model for phrase decoding. Although an alternative would be to use the IMT model to rescore a list of n-best lists from the HTR system, this decoupled approach has not been evaluated.

Finally, as it is typical in modern HTR and IMT models, the different probability distributions must be scaled. Here, the optimum language model scaling factor, λ, was chosen to optimize the average CER or WER in the development set of the three writers with the downhill simplex method [Nelder and Mead, 1965]. There were not significant differences in the optimum parameters obtained separately for each writer. Therefore, the estimation of these parameters seems rather robust to the variability of writers. The linear (or log-linear) interpolation factors for language model combination were also obtained using the simplex method over the development set.

5.6.1 Results on isolated words

The first scenario we have considered is to allow only the correction of a word at a time. Thus, the results must be interpreted as isolated word recognition. Figure 5.4 shows the test CER for different values of λ for the most relevant systems when recognizing isolated words. The plots are the true error curves which were obtained using the convex hull algorithm in a similar way to [Macherey et al., 2008; Tromble et al., 2008]. Circles ($\bullet$) indicate the optimum development λ. First, it must be noted that the optimum λ from the development set approximate quite well the test optimum, which is a desirable feature. The only exception is the 2**WG** system in Figure 5.4d for which an extra error reduction of 0.5 points could have been achieved.

Second, we should note the effect of adding **ERR** to the system on the error rate. A small improvement can be noticed in Spanish. However, the curves in English overlap. The explanation for this is a bit involving. Note that Spanish is a more inflected language than English. For example, *both* (in English) can be translated by *ambos* or *ambas* (in Spanish), depending on the gender, and having very similar writings. In contrast, *añade* (in Spanish) can be translated by *adds* (in English). Thus, we can see how translating from a less inflected language to a more inflected language introduces extra ambiguity. Furthermore, the possible translations of 'both' present also a similar spelling. Conversely, the ambiguity is reduced in the opposite direction. Table 5.1 shows the 5-best list of the HTR scores for the words *ambos* and *adds*. In the first case, *ambas* and *ambos* are the two most likely words in the HTR system, which differ in just one character and have similar HTR scores. Now, imagine that the IMT engine mistranslates *both* to *ambas*, by changing the gender of the word. Then, by saying that *ambas* is not correct with the **ERR** model, we give the system the opportunity to amend the error himself. However, in the English case, none of the words are synonyms of the word to recognize, and thus is more difficult to find the mistranslated word at the top of the n-best list. As a consequence, it is very unlikely that **ERR** achieves much improvement when translating from Spanish to English.

With respect to the n**PREF** models, only 4**PREF** has been displayed in the plots. The improvement over the baseline is consistent and significant. The

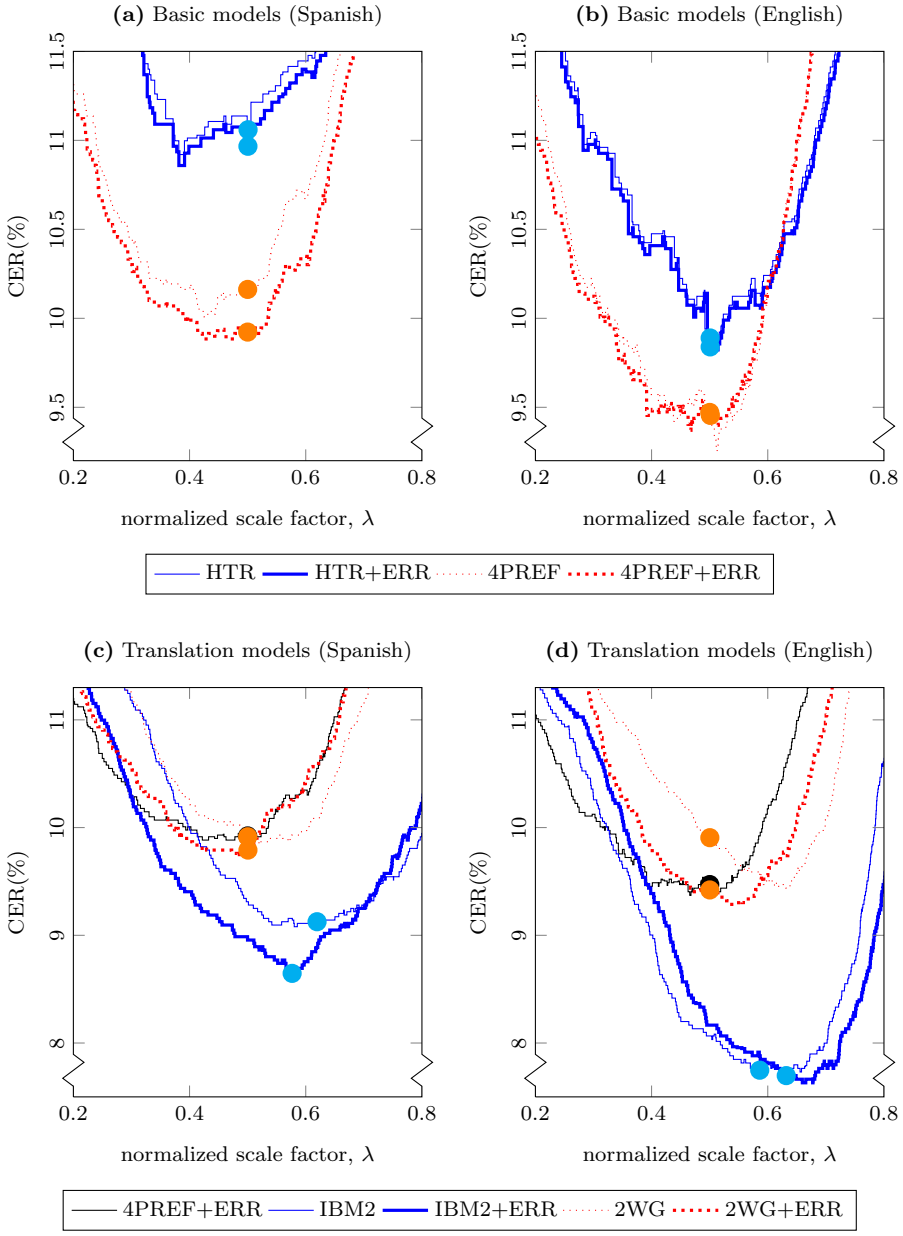


Figure 5.4: Test CER when modifying the λ scale factor. The x axis represents the variation of the normalized scale factor λ. The y axis shows the classification error rate (CER). Circles (•) indicate the optimum λ for the corresponding development sets. In the upper row, the comparison of the basic models. In the lower row, the most relevant translation models.

ambos		*adds*	
word	HTR score	word	HTR score
~~ambas~~	651.6	aids	137.9
ambos	*646.9*	cities	105.6
cambios	390.7	cycles	91.6
amplias	384.1	*adds*	*90.7*
campos	344.4	circles	85.8

Table 5.1: 5-best list for the words *ambos* and *adds*, which have been misrecognised. The ~~crossed-out~~ word is the word the IMT system mistranslated and the user is amending.

experiments were run on 2**PREF**, 3**PREF** and 5**PREF** as well. However, only 2**PREF** for English performed slightly worse than 4**PREF**. Longer prefixes achieved almost the same performance.

Regarding the systems using the translation models in Figure 5.4c and Figure 5.4d, we can see that these systems usually outperform the best basic system, 4**PREF+ERR**. The exception for this is 2**WG** for English, which shows a small performance degradation with respect to 4**PREF+ERR**. Still, 2**WG** systems do not seem to improve the basic systems significantly. Although several n**WG** systems were tested, any of them showed improvements over 2**WG**. On the other hand, **M2** systems achieve good improvements, although they are simpler than 2**WG**. A reason for that is that **M2** models have a smoother distribution probability and n**WG** systems need some sort of hypothesis pruning. In fact, the average number of candidates with probability greater than zero is 292 for **M2** and 38 for 4**WG**. **IMT** suffer even more from this problem with 2 candidates average.

A summary of the different alternatives studied for the word-based experiments is shown in Table 5.2. First, with only the basic information, 4**PREF+ERR** clearly outperforms **HTR**. Second, using translation models we can achieve further improvements. Since **M2** performs much better than **M1** we can deduce that alignment information is crucial for the translation models. Although inverse dictionaries have a better vocabulary coverage (4.7% vs 8.9% in English, 7.4% vs 10.4% in Spanish), they tend to perform worse than their direct dictionary counterparts. Still, inverse word models perform better than the n-grams alone. On the other hand, n**WG** performance is worse than word-based translation models. As it has been explained before, that might be due to the poorly smoothed probability distribution. Another reason might be that, in the process of obtaining n-gram models, information regarding alignments is lost as a result of the n-gram assumptions. When interpolating with 4**PREF**, **M2** models do not show significant improvements. In fact, for Spanish, the system presents over-fitting, since performance in development improves but in test decreases. However, 4**PREF** smooths 2**WG** distribution achieving close results

to word-based models. Moreover, a log-linear combination of the models show a bit of improvement with respect to word models alone. Nevertheless, linear interpolated models perform better. Other log-linear combinations (including a combination of all models) were tested. Nevertheless, none of them outperformed their linear interpolation counterpart. Next, by introducing **IMT**, small improvements can be obtained. Not surprisingly, **IMT** suffers from the same problems than n**WG**, but even more prominent. Finally, including all systems we can observe the best results overall, except for the over-fitting in the Spanish test set. Thus, 2**WG** seems to contribute slightly to improve the final model accuracy.

System	Spanish		English	
	dev	test	dev	test
HTR	9.7	11.1	7.9	9.9
ERR	9.6	11.0	7.8	9.8
4**PREF**	7.9	10.2	6.7	9.5
4**PREF+ERR**	7.8	9.9	6.6	9.5
2**WG+ERR**	8.6	9.8	7.7	9.4
M1-INV+ERR	8.4	9.5	7.5	9.2
M2-INV+ERR	7.9	9.1	7.1	9.1
M1+ERR	7.7	9.4	7.3	9.0
M2+ERR	7.1	**8.6**	5.9	7.7
2**WG+**4**PREF+ERR**	7.4	9.2	6.0	7.9
M2+4**PREF+ERR** (log-linear)	7.0	9.1	6.0	7.9
M2+4**PREF+ERR**	6.8	9.0	5.7	7.5
2**WG+**4**PREF+ERR+IMT**	7.3	9.2	6.0	7.9
M2+4**PREF+ERR+IMT**	**6.7**	8.9	5.7	7.5
ALL	**6.7**	8.9	**5.6**	**7.4**

Table 5.2: Summary of CER results for isolated word recognition. In this case, the user is allowed to amend one error at a time. The results show various language modeling approaches. In **boldface** the best systems.

5.6.2 Results on consecutive erroneous words

In contrast to the previous experiments, here we allow the correction of consecutive erroneous words. In this case, the decoding is performed as in continuous handwriting recognition. The results are summarized in Table 5.3. First, it should be pointed out that a third column has been added, which contains the best result achievable in the test sets when optimizing the parameters with the simplex algorithm instead of relying on the parameters from the development

set. This column is a lower bound of the error we can expect from the proposed models. It can be seen that the differences with respect to using the optimum development parameters are small in general. Thus, the estimation of the parameters is quite robust. First, it must be noted that the results for **ERR**, **M2**, and **IMT** are not shown, since they would require a different search engine. In addition, it is worth of mention that the baselines for phrase-based HTR have almost the double error rate than the word-based baselines. This is caused primarily because the segmentation for the words in the phrases are unknown. Then, it is the search algorithm that must find the most likely segmentation. As a result, segmentation errors are propagated to word errors. If we look at the results regarding the n**WG** models, they perform unexpectedly bad when used alone. However, when interpolated with 3**PREF** they show a good improvement. As in word-based recognition, word-based translation models show the best results, specially when interpolated with other models.

System	Spanish			English		
	dev	test	test*	dev	test	test*
HTR	15.9	16.8	16.7	13.0	18.6	18.4
3**PREF**	14.4	16.3	16.3	12.0	18.0	17.8
2**WG**	15.8	18.9	18.6	14.3	19.7	19.2
M1	14.2	17.0	16.8	12.2	17.4	17.4
2**WG**+3**PREF**	13.9	16.2	15.9	11.5	16.6	16.5
M1+3**PREF**	**12.6**	**15.2**	15.1	11.5	**15.5**	15.1
M1+2**WG**+3**PREF**	**12.6**	**15.2**	15.1	**11.1**	15.7	15.3

Table 5.3: Summary of WER results for continuous word recognition. In this case, the user is allowed to amend one or more consecutive errors in each interaction. The results show various language modeling approaches for the dev and test sets. Also, test* shows a lower bound if downhill simplex is used over the test set. In **boldface** the best systems.

To sum up, all the proposed systems significantly outperform the baseline recognizer. Basic models obtain a good improvement over the baseline. However, adding information from the translation may achieve remarkable results. Although more complex translation models suffer from smoothing problems, they can also contribute when interpolated with the rest of the models.

5.6.3 Error Analysis

An analysis (Table 5.4) of the results for the best word-based model shows that 49.2% to 54.4% of the recognition errors were produced by punctuation and other symbols. To circumvent this problem, we proposed a contextual menu in [Alabau et al., 2011]. With such menu, errors would have been reduced (best test result) to 4.4% in Spanish and 3.5% in English. Out-of-vocabulary

(OOV) words plus zero probability (P0) words (the words for which the decoder assigned zero probability or were pruned out) also summed up a big percentage of the error (40.3% and 28.9%, respectively). Finally, the rest of the errors were mostly due to one-to-three letter words, which can be basically a problem of handwriting morphological modeling.

class	words	word-based es (%)	word-based en (%)	phrase-based es (%)	phrase-based en (%)
punct.	., ,, :, ;, *, (,), —	49.2	54.4	14.0	18.6
1-char	a, e, y, o, u	4.1	0.9	8.3	2.3
2-char	of, if, la, by, on, is, ...	1.8	7.1	4.4	3.4
3-char	for, off, los, may, ...	0.0	4.3	2.1	4.9
numbers	xxvii, xxvi, xxiii, ...	2.3	0.9	2.1	2.3
OOV + P0	termina, luz, ...	40.3	28.7	20.2	13.6
others	latin, flash, fsma, ...	2.3	3.4	20.3	18.6
substitutions		100	100	71.5	63.8
insertions		–	–	3.0	4.6
deletions		–	–	25.5	31.6

Table 5.4: Detailed analysis of the word-based and phrase-based recognition errors. Five classes have been identified to produce the most amount of recognition errors. The second column shows samples of misrecognized words for these classes. Columns three and four are the percentage of these classes among the total number of misrecognized words for Spanish (es) and English (en), respectively. Columns five and six are the percentages for the phrase-based experiments. In this case, the percentage of substitutions, insertions and deletions is also shown.

On the other hand, phrase recognition presents a different error distribution. First, note that two new classes of errors have been introduced: deletions and insertions. The former account for the words in the reference that have been omitted, whereas the latter account for words inserted in the output hypothesis but do not correspond to any word in the reference. Both contribute to generate hypotheses with lengths different to their respective references, since the HMM models is not able to perform an accurate segmentation. Then, as a result, the proportion of recognition errors from the 'others' category increases from 3 to 20. In contrast, the proportion of errors regarding punctuation symbols decreases. Finally, it is to be remarked how the errors for short words have increased, probably because of small insertions or deletions.

5.6.4 Reducing Effort Correcting HTR errors

In case an HTR error is committed, the user may fall back to the virtual keyboard and type the correct word. The problem with this kind of keyboards is that typing is slow. To minimize this problem, we propose a contextual menu with a list of the n-best candidates (excluding the erroneous word). The aim is

to reduce the number of clicks needed to obtain the correct word with respect to a conventional virtual keyboard. As a baseline, for each HTR mistake, we count the number of clicks needed to input the correct word as: one click to pop up the keyboard, plus the number of characters in the word, plus one click to close the keyboard. For the Spanish test set, the average number of clicks per word amounts to 9.3, while for English it is 9.1 for the best word-based models in Table 5.2. This values can be surprisingly high, since it is known that the average word length is 4.5, i.e. the average number of clicks per word 6.5. However, it must be noticed that longer words are also more difficult to recognize. Thus, the average word length in the erroneous words is higher.

If the contextual menu is used, we count: one click for opening the menu plus one for choosing a word. If the correct word cannot be found in the n-best list, then we add: one count for the keyboard, plus the number of characters, plus a closing click. In Figure 5.5, we can see, on the left axis, the CER for a given size of the n-best list. Clearly, the error almost reduces to a quarter, around $n = 5$, with respect to the baseline. Between 10 and 15, the error stabilizes. Note that from 5 to 10 is still a reasonable amount of candidates to be shown in a circular menu. For more than 15, the CER almost equals the error for OOV+P0, since they cannot be found in n-best lists. On the right axis, we can observe the average number of clicks per word necessary to correct the mistakes. For $n = 1$ the number of clicks is reduced to 2.0. A trade-off can be found at $n = 7$ with 1.83 (80% relative improvement w.r.t. the baseline) and 1.82 (78% relative improvement), for Spanish and English, whereas the lower bounds are 1.75 and 1.73, respectively.

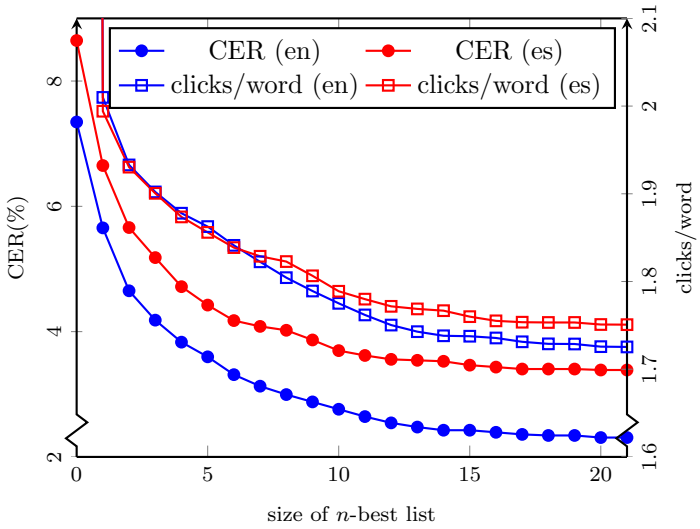


Figure 5.5: Reduction of CER and number of clicks as a function of the n-best list size.

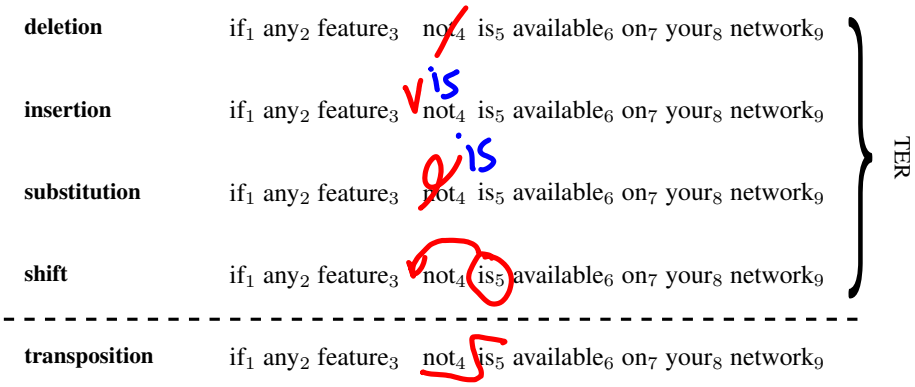


Figure 5.6: Illustration of the proof reading gestures devised for MT post-editing (marked as TER), and IMT (which comprises the whole set of gestures).

5.7 E-pen gestures

So far, e-pen interaction has been approached as just introducing the correct words. However, additional ways of interaction can be achieve by means of e-pen gestures. For instance, there is already a 'de facto' standard for gestures for proof reading from which we have extracted the most promising gestures (cf. Figure 5.6). This notation has been successfully used for years, although now that documentation is managed mainly in digital format, this technique is becoming obsolete. Nevertheless, we can convert such mature notation into gestures that can be understood by an interactive system. Then, the set of gestures that fit our post-editing needs are: substitutions, deletions, insertions and, transpositions. Furthermore, we have added a shift gesture to move phrases to specific places in the text (i.e., the user circles the phrase and draws an arrow to the final destination). Then, we have studied two e-pen post-editing approaches. In the first one, we consider substitutions, deletions, insertions and, shifts. The number of these operations to obtain a reference can be computed with the translation error rate (TER) [Snover et al., 2006]. In the second approach, we assume that the user is working with an interactive MT system (IMT). In this case, we have also considered transpositions.

To know what gestures could be more useful, we have conducted an experiment on the Xerox corpus (cf. Section 2.2.4). The summary of the edit rate results is displayed in Table 5.5. The edit rate is the number of edit operations needed to obtain the reference normalized by the number of words. We can see that the IMT system requires less interactions, especially for Spanish to English translation. Next, the number of times a particular edit operation has been applied is shown. We expect the gestures for deletion, insertion, shifting and transposition to be easy to tell apart for a machine learning algorithm. However, this will be the subject of future work. In addition, substitutions or

	post-editing		IMT	
	en-es	es-en	en-es	es-en
edit rate (%)	21.3	24.4	21.1	22.8
substitutions	1028	919	1549	1284
insertions	325	461	190	212
deletions	484	302	0	0
transpositions	–	–	41	56
shifts	319	357	347	354

Table 5.5: Summary of number of edit operations needed to obtain the reference for post-editing and interactive-predictive machine translation. The edit rate is the ratio between the number of edit operations and the number of words in the reference. Follows the number of occurrences for each edit operation. Here, we assume a perfect gesture recognizer. The gesture recognizer will be developed in future work.

insertions require the user to write the correct word, which can be done with a virtual keyboard or by handwriting, as we have seen in the previous sections.

5.8 Summary of contributions

In this chapter we have described a task specific on-line HTR system to operate with an IMT application. We have shown that a tight integration of the HTR and IMT decoding process can produce significant HTR error reductions. It is worth of note that all the proposed systems significantly outperform the baseline recognizer. Basic models obtain a good improvement over the baseline. However, translation models achieve remarkable results. Although more complex translation models suffer from smoothing problems, they also contribute when interpolated with the rest of the models. We also have introduced a new method for correcting HTR mistakes that consists on a contextual menu with the n-best candidates. The results show that a list with as few as 7 candidates allows to correct the HTR mistakes with just 1.83 clicks per word. Additionally, we have realized an initial exploration of which post-editing gestures are more likely to be useful.

On the other hand, the analysis of the results has shown two important issues to be tackled. First, the system should be able to decode unknown words since they are a clear limitation to system performance. A solution for this might be to use character language models instead of word language models, a technique that has achieved promising results in other areas [Zamora-Martínez et al., 2010]. Second, phrase-based models could benefit from better smoothing methods. Alignment information should be also taken into account more explicitly in these models. Furthermore, other alternatives could also be explored, as more advanced word-based translation models (such as HMM) that

cannot be used as n-grams in phrase-based decoding. These models could be used instead in the rescoring of the HTR WGs.

The work on HTR and IMT integration has lead to the following publications:

- **V. Alabau**, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition in Interactive Machine Translation. *Pattern Recognition*, submitted to(–), 2013.
- **V. Alabau**, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*, p. 389–394, 2011
- **V. Alabau**, D. Ortiz-Martínez, A. Sanchis, and F. Casacuberta. Multimodal Interactive Machine Translation. In *Proc. of the International Conference on Multimodal Interfaces (ICMI-MLMI'10)*, p. 46:1–46:4, 2010.

Additionally, the study of e-pen commands for MT and IMT was published in:

- **V. Alabau** and F. Casacuberta. Study of Electronic Pen Commands for Interactive-Predictive Machine Translation. In *International Workshop on Expertise in Translation and Post-editing Research and Application*, 2012.

Bibliography

V. Alabau, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*, p. 389–394, 2011.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J. Vilar. Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1):3–28, 2009.

A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A Maximum Entropy approach to Natural Language Processing. *Computational Linguistics*, 22:39–71, 1996.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The Mathematics of Machine Translation. *Computational Linguistics*, 19(2):263–311, 1993.

W. Campbell and F. Richardson. Discriminative Keyword Selection Using Support Vector Machines. In *Proc. of Advances in Neural Information Processing Systems (NIPS'07)*, p. 209–216, 2008.

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.

F. Farooq, D. Jose, and V. Govindaraju. Phrase-based correction model for improving handwriting recognition accuracies. *Pattern Recognition*, 42(12):3271–3277, 2009.

G. Foster, P. Isabelle, and P. Plamondon. Target-Text Mediated Interactive Machine Translation. *Machine Translation*, 12:175–194, 1998.

A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:855–868, 2009.

S. Green, J. Heer, and C. D. Manning. The efficacy of human post-editing for language translation. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*, p. 439–448, 2013.

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.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35 (3):400–401, 1987.

S. Khadivi and H. Ney. Integration of Speech Recognition and Machine Translation in Computer-Assisted Translation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1551–1564, 2008.

P. Koehn. *Statistical machine translation*, volume 11. Cambridge University Press, 2010.

P. Koehn and B. Haddow. Interactive Assistance to Human Translators using Statistical Machine Translation Methods. In *Proc. of MT Summit XII*, p. 73–80, 2009.

M. Liwicki and H. Bunke. HMM-Based On-Line Recognition of Handwritten Whiteboard Notes. In *Proc. of Tenth International Workshop on Frontiers in Handwriting Recognition (IWFHR'10)*, 2006.

W. Macherey, F. J. Och, I. Thayer, and J. Uszkoreit. Lattice-based minimum error rate training for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, p. 725–734, 2008.

J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965.

H. Ney, U. Essen, and R. Kneser. On the estimation of 'small' probabilities by leaving-one-out. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1202–1212, 1995.

F. J. Och and H. Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, p. 295–302, 2002.

K. Papineni, S. Roukos, and T. Ward. Maximum likelihood and discriminative training of direct translation models. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, p. 189–192, 1998.

M. Pastor and R. Paredes. Bi-modal Handwritten Text Recognition ICPR'10 Contest report. In *Proc. of the 20th International Conference on Pattern Recognition (ICPR'10)*, 2010.

M. Pastor, A. Toselli, and E. Vidal. Writing speed normalization for on-line handwritten text recognition. In *Proc. of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, volume 2, p. 1131–1135, 2005.

M. Plitt and F. Masselot. A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics*, 93(1):7–16, 2010.

S. Quiniou, M. Cheriet, and E. Anquetil. Error handling approach using characterization and correction steps for handwritten document analysis. *International Journal on Document Analysis and Recognition*, 15:1–17, 2011.

A. Sanchis, A. Juan, and E. Vidal. A Word-Based Naïve Bayes Classifier for Confidence Estimation in Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):565–574, 2012.

G. Sanchis-Trilles, D. Ortiz-Martínez, J. Civera, F. Casacuberta, E. Vidal, and H. Hoang. Improving Interactive Machine Translation via Mouse Actions. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, p. 485–494, 2008.

M. Shilman, D. S. Tan, and P. Simard. CueTIP: a mixed-initiative interface for correcting handwriting errors. In *Proc. of the 19th annual ACM symposium on User interface software and technology (UIST'06)*, p. 323–332, 2006.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. of Proceedings of association for machine translation in the Americas (AMTA'06)*, p. 223–231, 2006.

B. Suhm, B. Myers, and A. Waibel. Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction*, 8(1):60–98, 2001.

A. H. Toselli, V. Romero, M. Pastor, and E. Vidal. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825, 2010.

R. W. Tromble, S. Kumar, F. Och, and W. Macherey. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, p. 620–629, 2008.

N. Ueffing, F. Och, and H. Ney. Generation of word graphs in statistical machine translation. In *Proc. of the Conference on Empirical methods in natural language processing (EMNLP'02)*, p. 156–163, 2002.

F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.

F. Zamora-Martínez, M. J. Castro-Bleda, S. España-Boquera, and J. Gorbe-Moya. Unconstrained offline handwriting recognition using connectionist character N-grams. In *Proc. of the International Joint Conference on Neural Networks (IJCNN'10)*, p. 1–7, 2010.

R. Zens, F. Och, and H. Ney. Phrase-based statistical machine translation. In *Proc. of the 25th Annual German Conference on Artificial Intelligence (KI'02)*, p. 35–56, 2002.

Chapter 6

Speech Interaction for Translation and Handwritten Text Transcription

Chapter Outline

6.1 Introduction

In the previous chapter, HTR performance has been improved by using specific information from the task in which this interaction modality was being applied. In this chapter, similar ideas are employed to improve *automatic speech recognition* (ASR) when used as an interaction modality for translation and handwritten text transcription.

Speech interaction can be conceived as a natural modality to improve system ergonomics and productivity. For instance, Dragsted et al. [2011] performed an study where translators where asked to translate short texts in three conditions: written translation, sight translation without ASR[1], and sight translation with an ASR tool. Although with written translation the quality achieved was slightly better, sight translation was three times more productive, i.e., it took the participants a third of the time to complete the task. However, sight translation with ASR took more than twice the time with respect to plain sight translation, since the participants had to review and post-edit the ASR output. Additionally, when consulting informally paleographists on the most comfortable method to transcribe a handwritten text document, many of them claim that a dictation of the words would be an interesting choice. Consequently, speech interaction with off-line *handwritten text recognition* (HTR) systems should also be taken into consideration.

As it happened with e-pen interaction in Chapter 5, the problem to face with this alternative modality is that it is not deterministic, i.e., speech needs to be decoded by a specific system which may commit errors. Since the user would seldom use an error prone system, their utility is conditioned to high accuracy rates. Thus, one of the most important research problems in this area is how to take advantage of the information available in the structured prediction problem being tackled to improve the final ASR accuracy. This leads to an interesting and challenging fusion problem. The concurrent multimodal aspects of the interaction process, i.e. how keyboard, mouse and the alternative modalities could be used simultaneously, will be left apart as they occur in rare occasions [Oviatt and VanGent, 1996]. On the contrary, it is assumed that the user will prefer to use the alternative modality as long as the recognition accuracy is high enough, if she finds the modality to be more productive or ergonomic. The user will fall back to the keyboard just in case of a decoding mistake [Shilman et al., 2006].

Several techniques for dictating translations were explored in the nineties, for instance by Brown et al. [1994] and Dymetman et al. [1994], but this topic has received some attention recently. Paulik et al. [2005] used a *cache* language model with the uni-grams obtained from an MT n-best list. On the

[1] In sight translation, the translator or interpreter is given a written document in the source language and is asked to read it aloud in the target language.

other hand, Reddy and Rose [2010] approached sight dictation by combining speech and translation at phonetic level using an edit distance algorithm with a composition of weighted finite-state automata. However, the first study of a speech-enabled interactive system was Vidal et al. [2006]. There, several scenarios were proposed, where the user was expected to speak aloud parts of the current hypothesis and possibly one or more corrections. The technique consisted on rescoring the language model with the probabilities of word-based translation models [Brown et al., 1993], which are not aware of the context of the source and target sentence. Latter, several methods were proposed in [Khadivi and Ney, 2008] that took advantage of context information. *Word graphs* (WGs) from speech and translation were combined in different ways including speech and translation WG composition using finite-state automata toolkits and WG rescoring with word-based translation models and posterior n-grams from m-best lists. However, the approaches presented in that work has one main drawback: the fusion problem was approached as late fusion as the ASR WG was generated without taking into account MT information. Regarding dictation in HTR, previous attempts in combining handwritten input and speech input have been done [Liu and Soong, 2009], but most of them focus on the use of on-line handwritten text.

This chapter is devoted to exploring new techniques to provide a more reliable speech-enabled input interface that can lead to a real multimodal system. In particular, we focus on approaches that can be seamlessly integrated with current speech recognition technology. The techniques we propose are very similar to the ones explored in Chapter 5, which allow a context aware decoding. We have considered two scenarios. The first one is a speech-enabled IMT system [Vidal et al., 2006] and the second one could be called *sight transcription*, since the user reads aloud the transcription of a handwritten document.

6.2 Speech-enabled IMT

Enabling a speech interface for MT is much in the line of enabling on-line HTR for MT (cf. Chapter 5). Only that in this case, the variable $\boldsymbol{f}$ represents a speech signal instead of a sequence of pen strokes. Nevertheless, in speech it makes less sense to force the user to utter single words. Thus, only the scenario where multiple words can be spoken will be analyzed. Additionally, we will use the modeling strategies analogous to those that were used in Chapter 5, with the goal to find solutions compatible with existing speech technologies and decoders.

As a reminder, an illustration of a speech-enabled IMT system is shown in Figure 6.1. To distinguish when keyboard or speech is used, we will denote κ to represent a word (or sequence of words) introduced by using the keyboard, and $\boldsymbol{f}$ to represent a speech utterance to be decoded. As the example shows, the process starts with an empty prefix $\boldsymbol{y}_p$, so the system proposes in the first iteration a full translation $\hat{\boldsymbol{y}}$. This output would be the same of a conventional

MT system. Then, the user selects the error-free prefix $\boldsymbol{y}_p$ and speaks aloud to correct the first translation error. As the output of the ASR system, $\hat{\boldsymbol{d}}$, is wrong, the user falls back to using the keyboard, κ. Then, the system proposes a new suffix $\hat{\boldsymbol{y}}_s$ based on the user feedback. In the second iteration, the user amends *at* by uttering the word *in* and the system reacts by predicting a new suffix $\hat{\boldsymbol{y}}_s$. Since this suffix is fully correct the process ends. Note that the speech-enabled IMT approach has allowed to obtain the correct translation with only two user corrections, whereas more effort would have been required if the conventional approach, based on fully automatic MT and human post-editing had been applied.

SOURCE ($\boldsymbol{x}$): si alguna función no se encuentra disponible en su red
REFERENCE ($\boldsymbol{r}$): if any feature is not available in your network

ITER-0	$(\hat{\boldsymbol{y}})$	if any feature not is available on your network
ITER-1	$(\boldsymbol{y}_p)$	*if any feature*
	$(\boldsymbol{f})$	
	$(\hat{\boldsymbol{d}})$	~~in~~
	(κ)	`is`
	$(\hat{\boldsymbol{y}}_s)$	not available at your network
ITER-2	$(\boldsymbol{y}_p)$	if any feature is *not available*
	$(\boldsymbol{f})$	
	$(\hat{\boldsymbol{d}})$	**in**
	$(\hat{\boldsymbol{y}}_s)$	your network
FINAL	$(\hat{\boldsymbol{y}} = \boldsymbol{r})$	if any feature is not available in your network

Figure 6.1: Example of an ASR-enabled IMT session for translating a Spanish sentence $\boldsymbol{x}$ from the Xerox corpus to an English sentence $\boldsymbol{r}$. In each iteration, the user selects the longest error-free prefix $\boldsymbol{y}_p$, e.g., by positioning the mouse cursor. Then the user speaks aloud the correction, $\boldsymbol{f}$, which can be composed my one or more words. If the decoding of the utterance, $\hat{\boldsymbol{d}}$, is correct, then it is displayed in **boldface**. On the contrary, if $\hat{\boldsymbol{d}}$ is incorrect, it is shown ~~crossed out~~. In this case, the user amends the error using the virtual keyboard κ (in `typewriter`). Finally, the system proposes a new suffix ($\hat{\boldsymbol{y}}_s$) based on the user's feedback ($\hat{\boldsymbol{d}}$ or κ). This process continues until the reference, $\boldsymbol{r}$, is reached.

The IMT process starts producing a full translation $\hat{\boldsymbol{y}}$ of the source sentence $\boldsymbol{x}$ based on conventional statistical MT techniques. Then, according to the source sentence, the user analyzes $\hat{\boldsymbol{y}}$ in order to detect some possible translation errors. Typically, the user reads $\hat{\boldsymbol{y}}$ from left to right and the first error found is replaced by some text $\hat{\boldsymbol{d}}$ resulting from the user's dictation of the correction. Then, a *validated* error-free prefix $\boldsymbol{y}_p$,$\hat{\boldsymbol{d}}$ is generated. This prefix is used by the IMT system in the next iteration to produce a new prediction $\hat{\boldsymbol{y}}_s$ so that the concatenation of $\boldsymbol{y}_p$,$\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{y}}_s$ constitutes a whole translation of the source sentence. This process is repeated until a completely error-free translation is achieved. Alternatively, the user can use the keyboard κ.

In our strategy, the user's feedback is aimed at introducing a correction after the correct prefix. However, as opposed to the HTR scenario, the system does not know where to introduce the result from the feedback. In this work, we will assume that the user has positioned the cursor in the exact place where the result is supposed to appear. However, an alternative would be validating a fragment of the suggestion (prefix selection, $\boldsymbol{y}_p$) by using speech as well. This should a very easy task since it is restricted to utter a fragment of the current IMT prediction, and thus, the perplexity would be very low. In fact, Vidal et al. [2006] achieved word error rates for prefix selection around 1.6% for the same task analyzed in this chapter. Around 96.4% of the utterances were perfectly recognized, which suggest that the prefix selection scenario is almost usable following LaLomia [1994] recommendations. In addition, gaze-tracking could also be used to set the cursor position. In that case, the user should just start speaking while looking at the spot where the cursor should be.

This section is organized as follows. First, a brief survey of the techniques proposed by other authors is presented in Section 6.2.1. Next, Section 6.2.2 describes our contribution for the integration of the MT context in the speech recognizer. As these techniques are very similar to the ones explored in Chapter 5 we will only make a brief overview. For more details on any of these techniques, refer to Chapter 5.

6.2.1 Leveraging task-specific context

If we were to use a completely decoupled ASR system to enable speech multimodality in MT, the optimal decoding could be obtained as

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}) = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d}) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{6.1}$$

where $\Pr(\boldsymbol{d})$ denotes the *language model*, which deals with the probability of $\boldsymbol{d}$ being a sequence of words in a specific language. $\Pr(\boldsymbol{f} \mid \boldsymbol{d})$ is the *acoustic model*, which is a representation of the probability distribution for the constituent speech elements (usually phonemes or phoneme sequences) in the input utterance. For more details regarding how ASR modeled see Section 2.2.3.

This first approximation, referred as DEC in [Vidal et al., 2006], is based on the use of a conventional ASR system to obtain $\hat{\boldsymbol{d}}$. Equation (6.1) can be regarded as a completely independent module (black-box) and, therefore, it could be adopted to build a speech input interface for any kind of interactive system. However, in MT, we can benefit from the context provided by the interactive system itself to introduce speech recognition in such a way that the speech decoding accuracy can be improved. It is reasonable to assume that the user will utter something that is both a suitable continuation to the current selected prefix and a fragment of a correct translation for the source sentence. This fact can be profitably used to actually improve the ASR performance.

With this purpose, Vidal et al. [2006] proposed a new alternative by introducing the prefix $\boldsymbol{y}_p$ as a language model constrain in Equation (6.1):

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}, \boldsymbol{y}_p) \tag{6.2}$$

Under the assumption that $\Pr(\boldsymbol{f} \mid \boldsymbol{d}, \boldsymbol{y}_p)$ does not depend on $\boldsymbol{y}_p$ if $\boldsymbol{d}$ is known, Equation (6.2) can be rewritten as

$$\hat{\boldsymbol{d}} \approx \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{y}_p) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{6.3}$$

Here, the language model is conditioned to the current prefix. Consequently, the search space in the ASR decoding can be constrained to those hypotheses that are suitable continuations of $\boldsymbol{y}_p$. Note that Equation (6.3) is the same approach described for HTR in Equation (5.6).

Similarly, the source sentence can be introduced into the language model in Equation (6.1) as a constraint:

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}, \boldsymbol{x}) \tag{6.4}$$

and, under the assumption that $\Pr(\boldsymbol{f} \mid \boldsymbol{d}, \boldsymbol{x})$ does not depend on $\boldsymbol{x}$ if $\boldsymbol{d}$ is known, we can approximate Equation (6.4) by

$$\hat{\boldsymbol{d}} \approx \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{x}) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{6.5}$$

Now, this constraint favors decodings of $\hat{\boldsymbol{d}}$ whose words result from translations of $\boldsymbol{x}$. In this case, the information on the target sentence context is not taken into account.

Clearly, a more interesting and challenging approach is to consider dependencies on both $\boldsymbol{x}$ and $\boldsymbol{y}_p$:

$$\hat{\boldsymbol{d}} = \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{f}, \boldsymbol{y}_p, \boldsymbol{x}) \tag{6.6}$$

If we assume that $\Pr(\boldsymbol{f} \mid \boldsymbol{d}, \boldsymbol{y}_p, \boldsymbol{x})$ does not depend on $\boldsymbol{y}_p$ and $\boldsymbol{x}$ if $\boldsymbol{d}$ is known, then Equation (6.6) can be rewritten as

$$\hat{\boldsymbol{d}} \approx \arg\max_{\boldsymbol{d}} \Pr(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x}) \Pr(\boldsymbol{f} \mid \boldsymbol{d}) \tag{6.7}$$

which is equivalent to the Equation (5.9) used in the case of HTR.

The use of Equation (6.7) has been addressed in previous works [Khadivi and Ney, 2008; Rodríguez-Ruiz, 2010; Vidal et al., 2006]. In [Vidal et al., 2006] word-based translation models were used to rescore the ASR language model.

Later, [Rodríguez-Ruiz, 2010] rescored n-best list hypotheses from the ASR output with word-based models. However, in word-based models the word probabilities of the target sentence are independent from each other, and thus contextual information is not used. This is a known limitation, since translation models using contextual information as n-gram-based [Casacuberta et al., 2005; Mariño et al., 2006] or phrase-based [Och and Ney, 2002] models greatly outperform word-based models.

Although phrase-based WGs did not improve word-based models in HTR decoding in Chapter 5, previous works from other authors indicate that probably we could obtain bigger benefits in the case of ASR. For instance, the techniques in [Khadivi and Ney, 2008] used a WG from an ASR system, as input to a series of combinations with word-based and phrase-based translation models. The results were promising. However, the problem of these approaches is that the best solution is limited by the quality of the ASR WG. A way of measuring such quality is by the error rate of the sentence in the WG closest to the reference, as it is a lower bound on the error rate. Rescoring the ASR WG cannot recover from errors in the ASR stage. Hence, an early fusion of the translation models would allow to recover some of these errors otherwise impossible to amend. Additionally, Khadivi and Ney [2008] experiments were more similar to the *sight translation* in [Reddy and Rose, 2010] than to an speech-enabled IMT scenario.

6.2.2 Speech and translation fusion

Typically, state-of-the-art ASR systems approach the language modeling with n-gram models [Jelinek, 1998]. Therefore, if the constraints could be encoded as an n-gram language model, the integration with current state-of-the-art ASR systems would become trivial. A simple way to do so is to transform word-based translation models into a n-gram-style language model to model Equation (6.5). Thus, the probability of the translation model can be decomposed based on a strong naïve Bayes assumption similar to the assumptions for the word-based statistical dictionaries [Brown et al., 1993],

$$Pr(\boldsymbol{d} \mid \boldsymbol{x}) = \prod_k \Pr(d_k \mid d_1^{k-1}, \boldsymbol{x}) \approx \prod_k p(d_k \mid \boldsymbol{x}) \tag{6.8}$$

Then, the probability of a target word being a translation of the source sentence can be computed similarly to Equation (5.12) by

$$p(d_k \mid \boldsymbol{x}) \quad = \quad \sum_{j=1}^{|\boldsymbol{x}|} p(k \mid \boldsymbol{x})\, p(d_k \mid j, \boldsymbol{x}) \tag{6.9}$$

where $p(j \mid \boldsymbol{x})$ is an alignment model which can be approximated, as in M1, by $\frac{1}{|\boldsymbol{x}|+1}$ and, $p(d_k \mid j, \boldsymbol{x})$ is a dictionary which can be approximated by a statistical

lexicon $p(d_k \mid x_j)$ models [Brown et al., 1993]. Hence, Equation (6.9) can be approximated by

$$p(d_k \mid \boldsymbol{x}) \quad \approx \quad \sum_{j=1}^{|\boldsymbol{x}|} \frac{p(d_k \mid x_j)}{|\boldsymbol{x}|+1} \tag{6.10}$$

As we explained in Section 5.6, M1 models can be easily encoded as a 1-gram language model, which is very convenient to integrate the model into a state-of-the-art ASR system. Unluckily, M2 models cannot be so easily integrated when the output may consist in more than one word. As that is the case for a speech-enabled interface, we have decided not to use M2 in this section.

On the other hand, to deal with the application of Equation (6.7), we can use state-of-the-art phrase-based model [Och and Ney, 2002] to build n-gram posterior probabilities over WGs generated by such systems (see Section 5.4.2 to see the details of this technique). As a result, a (smoothed) n-gram language model is obtained that is constrained to the source sentence and the correct prefix. The resulting language model can be defined as

$$p(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x}) = \prod_i p(d_i \mid d_{i-n+1}^{i-1}, \boldsymbol{y}_p, \boldsymbol{x}) \tag{6.11}$$

which needs to be smoothed to account for words not seen in the WG (cf. Section 5.4.2). A last approach to avoid poor estimations of $p(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x})$ is a linear interpolation with a regular language model,

$$p_\gamma(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x}) = \gamma\, p(\boldsymbol{d} \mid \boldsymbol{y}_p, \boldsymbol{x}) + (1-\gamma)\, p(\boldsymbol{d} \mid \boldsymbol{y}_p) \tag{6.12}$$

where γ is the interpolation factor.

6.2.3 Results of speech-enabled IMT

This section is devoted to analyzing the experimental results of the methods proposed in Sections 6.2.1 and 6.2.2. The experiments were carried out with the speech Xerox corpus described in Section 2.3.2. This corpus was obtained by retrieving consecutive errors from IMT sessions, which constitute the segments of text that the users would speak aloud. Thus, the results presented in this section refer to the decoding of such segments of texts, not to full sentences. In fact, this aspect makes speech interaction a harder problem that plain dictation, as reflected by 3-gram perplexities (163 vs 48, respectively). In order to build the word-based and phrase-based models required for the experiments, the publicly available toolkit for IMT, Thot[2] [Ortiz-Martínez et al., 2005], was used. With the word-based models, **M1** language models were generated as in Equation (6.9). In addition, the phrase-based models were used to generate the translation WGs for each source sentence in the test dataset. These WGs were

[2] http://sourceforge.net/projects/thot/

used to estimate the n-gram posterior models (n**WG**) in Equation (6.11). For the models needing parameter adjustment, as posterior scale or ϵ in absolute discount, a range of parameter values were explored. However, only the most promising models, in terms of perplexity, were selected for the experiments. The analysis of this will be shown later in this section. Next, the parameters of the speech recognizer were estimated based on WER over a held-out development dataset for the 3-gram language model. Finally, the baseline and selected models were used to generate ASR WGs with the publicly available iATROS[3] [Luján-Mares et al., 2008] speech recognition software. The quality of the models was then assessed by the perplexity of the model, the WER of the hypothesis with maximum probability, and the OWER, i.e. the hypothesis in the WG with minimum WER.

As a baseline, two scenarios, shown in Table 6.1, have been considered. The 3**GRAM** system in Equation (6.1), which uses an unconstrained language model, obtains a 26.7 of WER and a 12.0 of OWER. The 3**PREF** system in Equation (6.3), which is conditioned on $\boldsymbol{y}_p$, obtains 23.1 and 10.4, respectively. Note that, not only the performance is better, but the quality of the WGs also improves.

model	PPL	WER	OWER
3**GRAM**	163	26.7	12.0
3**PREF**	57	23.1	10.4

Table 6.1: Perplexity, WER and oracle WER (OWER) for the baseline system.

Conversely, Table 6.2 shows the results for the models not depending on $\boldsymbol{y}_p$, that is, independent of the target language context. The **M1** system proposed in this work is only conditioned on $\boldsymbol{x}$. Although perplexity results are better than the previous approaches (34 vs 57 of 3**PREF** and 163 of 3**GRAM**), WER results are quite worse (29.9) and the WG quality is bad. However, **M1** has no contextual information, so it is more similar to a 1**GRAM** system. In fact, it is encoded as such. In this comparison, the **M1** system outperforms the 1**GRAM** system. The last system of this category is 1**WG**. This is actually the one with more information, since it has been obtained from a phrase-based model including various sources of information from the log-linear translation model. Indeed, it shows the best performance compared to 1**GRAM** and **M1**. Moreover, it outperforms 3**GRAM** and is close to the 3**PREF** system. When these systems are combined with a 3**PREF** model to provide contextual information the results are comparable or better than the 3**PREF** model alone (with interpolation factor $\gamma = 0.6$ for **M1** and $\gamma = 0.8$ for 1**WG**). It is worthy of note the OWER results, that improve dramatically compared to non interpolated models. With respect to the 3**PREF** system, which would be the one

[3] http://prhlt.iti.upv.es/w/iatros

used in [Khadivi and Ney, 2008], interpolated models improve the WG quality at least a 27%. Hence, WG rescoring techniques could be able to improve results when applied to WGs generated using interpolated models.

model	PPL	WER	OWER
1**GRAM**	455	43.2	37.6
M1	34	29.9	26.2
1**WG**	27	24.7	21.5
M1 + 3**PREF**	22	23.2	8.7
1**WG** + 3**PREF**	28	22.4	8.5

Table 6.2: Perplexity, WER and oracle WER (OWER) for models independent from target language context ($\boldsymbol{y}_p$) and respective linear interpolation models. Note that 1**GRAM** and 1**PREF** are the same models since both lack of dependency on $\boldsymbol{y}_p$ or $\boldsymbol{x}$

The posterior scale parameter, that was defined in Equation (3.21), is crucial in word posterior probability estimation [Ogawa et al., 1998]. Typically, this parameter must be adjusted to balance the probabilities of the competing hypotheses. Figure 6.2 shows the results in perplexity n**WG** systems of order from 1 to 5 when varying the posterior scale. Unexpectedly, the best posterior scale is found at 1. However, the explanation for this is rather simple. The word posterior probabilities have been used mainly on ASR WGs in the literature. Here, the acoustic score is not actually a probability but a density. Hence, the scores are very low and the posterior scale is needed to obtain sensible word posterior probabilities. On the contrary, translation models are based on counts, and thus, the WGs encode actual probabilities. Then, it is not necessary to scale the scores to balance the competing hypotheses.

The n**WG** approach was tested for various orders of n-grams, from 1 to 5. The results for perplexity and WER are displayed on Figure 6.3. Diamonds ($\diamond$) represent n**WG** models while squares ($\square$) represent n**WG** models interpolated with a 3**PREF** model. First, note that perplexity and WER results overlap. This has been made on purpose to show the clear correlation in perplexity and WER for this kind of models. A minimum is obtained at 2**WG**. Then as the n-gram order increases, so does the error. This seems a symptom of model overfitting. However, all the systems outperform significantly the previous models for orders of 2 or more. Not surprisingly, these models have managed adequately the integration of the source sentence $\boldsymbol{x}$ and the target context $\boldsymbol{y}_p$. At last, when interpolating with n**PREF** models a consistent improvement is shown. In addition, this smoothing is able to overcome the problems of overfitting and, now, the performance stabilizes for higher order of n-grams.

Finally, Table 6.3 summarizes the experiments conducted on this chapter. Interpolation factors are $\gamma = 0.6$ for **M1** and $\gamma = 0.8$ for 3**WG**. As it can be observed, all proposed models outperform previous models in their respective

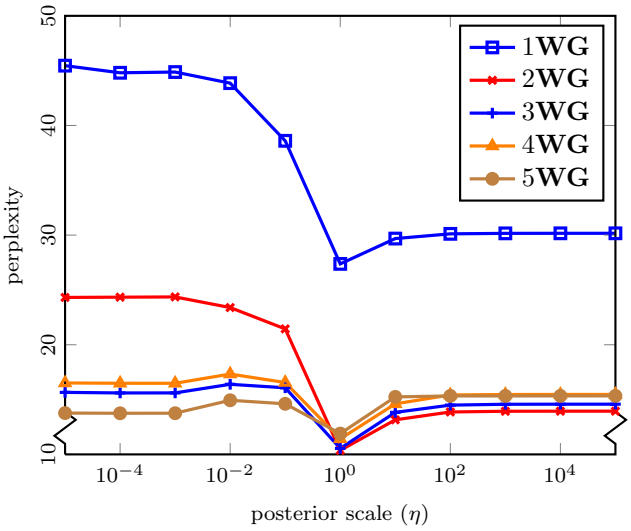


Figure 6.2: Perplexity for various posterior n-grams when varying the posterior scale.

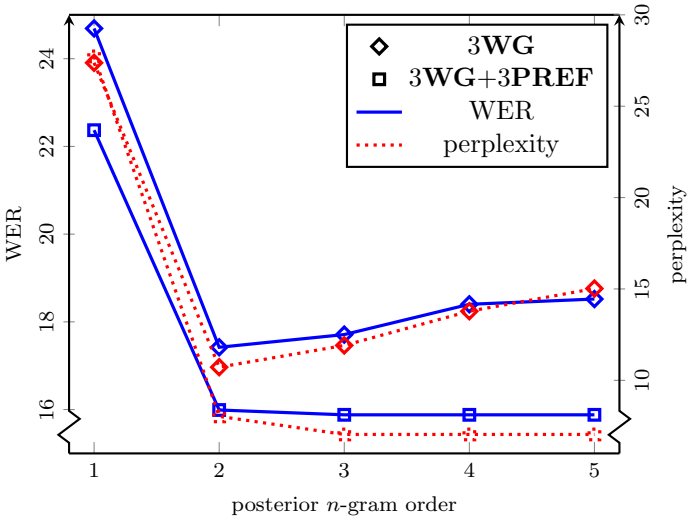


Figure 6.3: WER and perplexity (PPL) results for different orders of posterior n-grams.

categories. Besides, the generated WGs have also greater quality in terms of OWER improving in 44.2% relative with respect to 3**GRAM** for the best model. This would allow WG rescoring techniques to improve their results.

model	PPL	WER	OWER
3**GRAM**	163	26.7	12.0
3**PREF**	57	23.1	10.4
1**GRAM**	455	43.2	37.6
M1	34	29.9	26.2
1**WG**	27	24.7	21.5
M1 + 3**PREF**	22	23.2	8.7
2**WG**	11	17.7	8.9
3**WG** + 3**PREF**	**7**	**15.9**	**6.7**

Table 6.3: Summary of perplexity, WER and oracle WER (OWER) for the different approaches. Baseline results in the first block. The second block for alternatives not using $\boldsymbol{y}_p$. In the third block alternatives using $\boldsymbol{y}_p$ and $\boldsymbol{x}$.

6.3 Dictation of handwritten manuscripts

In previous sections, we have seen how HTR and ASR can be used to interact with an interactive machine translation system. The problem we are dealing here is a bit different since it is a dictation problem in a more similar way to the works of Brown et al. [1994] and Reddy and Rose [2010]. In this case, the original input and the user's feedback carry the same message, and hence, the decoding of both signals must coincide. Basically, the problem consists in obtaining a sequence of words $\boldsymbol{y}$ which is, at the same time, a transcription of the handwritten text image $\boldsymbol{x}$ (from the HTR problem) and a speech utterance $\boldsymbol{f} = (f_1, f_2, \ldots, f_{|\boldsymbol{f}|})$. Statistically, this problem can be formulated as

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{x}) \Pr(\boldsymbol{f} \mid \boldsymbol{x}, \boldsymbol{y}) \tag{6.13}$$

Making the safe assumption that $\Pr(\boldsymbol{f} \mid \boldsymbol{x}, \boldsymbol{y})$ is independent of $\boldsymbol{x}$ if $\boldsymbol{y}$ is known, Equation (6.13) can be rewritten as

$$\hat{\boldsymbol{y}} \approx \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{x}) \Pr(\boldsymbol{f} \mid \boldsymbol{y}) \tag{6.14}$$

where $\Pr(\boldsymbol{y} \mid \boldsymbol{x})$ is a language model conditioned on the handwritten text image, $\boldsymbol{x}$, and $\Pr(\boldsymbol{f} \mid \boldsymbol{y})$ is a conventional acoustic HMM for ASR. Note that if $\boldsymbol{x}$ is dropped, the language model can be approximated by a standard n-gram language model. In that case, Equation (6.14) can be decoded with a state-of-the-art ASR system. However, a more interesting approach would be to take advantage of the information given by $\boldsymbol{x}$.

Although in principle an integrated decoding could be possible [Bengio, 2004], it would require a specific training and decoding. This is especially complicated since both input signals have different lengths and are not synchronized. A

possible alternative is based on a semi-coupled approximation, in which $\Pr(\boldsymbol{y} \mid \boldsymbol{x})$ can be transformed into a statistical language model that can be used with current ASR systems. To generate such language model, we can use the same procedure as in Section 5.4.2 but from a WG generated by a HTR system.

Note that Equation (6.13) could have been decomposed, as well, in the following way,

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{f}) \Pr(\boldsymbol{x} \mid \boldsymbol{f}, \boldsymbol{y}) \tag{6.15}$$

By following the same assumptions than in Equation (6.13), we can rewrite Equation (6.15) as

$$\hat{\boldsymbol{y}} \approx \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y} \mid \boldsymbol{f}) \Pr(\boldsymbol{x} \mid \boldsymbol{y}) \tag{6.16}$$

Obviously, the estimation of $\Pr(\boldsymbol{y} \mid \boldsymbol{f})$ can be done by means of posterior n-grams from the ASR WG, as it was done with the HTR WG (cf. Section 5.4.2). Initially, we can find empirically which alternative of both Equation (6.14) or Equation (6.16) to use. However, the intuition says that the system with lower error rate would constitute a better prior for the system with higher error rate.

6.3.1 Iterative decoding

In contrast to our previous approaches to integrate speech and handwriting in IMT, where the utterances were fragments of the correct translations, in this case both the handwritten sentence and the utterance convey the very same text. Thus, as we can use any of the signals to obtain the n-gram models, it seems logical that an iterative procedure can be applied. Here, HTR and ASR WGs are used alternatively in order to improve the final result of the system, in a similar way to a dual optimization problem.

The algorithm for the iterative procedure is represented in Algorithm 2. Note that we have added a new input to the procedure, I, that represents the maximum number of iterations. This is necessary since there is no proof for convergence for this algorithm and, in practice, it does not converge in some specific cases.

6.3.2 Results of dictation of manuscripts

This section is devoted to analyze the experimental results for the "Cristo-Salvador" corpus defined in Section 2.3.3. We will study the methods proposed in Section 6.3 and the iterative process described in Section 6.3.1. The results will be compared against a baseline result which uses only one modality and in a non-iterative fashion.

Algorithm 2: Iterative decoding algorithm.

Input: $\boldsymbol{x}, \boldsymbol{f}, I$
Output: $\boldsymbol{y}$
$i \leftarrow 0$;
$\hat{\boldsymbol{y}}_s^{(0)} \leftarrow \arg\max_{\boldsymbol{y}} p(\boldsymbol{f} \mid \boldsymbol{y}) p(\boldsymbol{y} \mid \boldsymbol{x})$;
repeat
  $i \leftarrow i + 1$;
  $p^{(i)}(\boldsymbol{y} \mid \boldsymbol{f}) \leftarrow n$-gram posteriors from WG($\boldsymbol{f}$) in iteration $i - 1$;
  $\hat{\boldsymbol{y}}_{\boldsymbol{x}}^{(i)} \leftarrow \arg\max_{\boldsymbol{y}} p(\boldsymbol{x} \mid \boldsymbol{f}, \boldsymbol{y}) p^{(i)}(\boldsymbol{y} \mid \boldsymbol{f})$;
  $p^{(i)}(\boldsymbol{y} \mid \boldsymbol{x}) \leftarrow n$-gram posteriors from WG($\boldsymbol{x}$) in iteration i;
  $\hat{\boldsymbol{y}}_{\boldsymbol{f}}^{(i)} \leftarrow \arg\max_{\boldsymbol{y}} p(\boldsymbol{f} \mid \boldsymbol{x}, \boldsymbol{y}) p^{(i)}(\boldsymbol{y} \mid \boldsymbol{x})$;
until $\hat{\boldsymbol{y}}_s^{(i)} = \hat{\boldsymbol{y}}_s^{(i-1)}$ **or** $i = I$;
return $\hat{\boldsymbol{y}}_s^{(i)}$

Baseline results

First of all, it should be noted that, despite being a small corpus to what the speech community is used to, the corpus used here is a realistic example of what can be found in transcription of historical documents.

There are some characteristics of this kind of tasks that must be explained. On the one hand, the topic addressed is a very specific one. Since the training corpus is rather small ($6.4k$ running words), language models are poorly estimated. This is reflected in the perplexity for the test page (552 for a 2-gram). Higher order n-gram models cannot improve perplexity since segments longer than 2 words rarely occur more than once. Furthermore, as far as we know, there are no other electronic texts dealing with the same topic, and consequently no robust language models can be estimated. Other texts with the writing style of the nineteenth century are simply too different to be useful, e.g. most of them are literary texts. As a result, both HTR and ASR baseline systems must rely more on the good estimation of the HMM models. On the other hand, each book presents a particular handwriting style which not only depends on the author, but on the period of the history the book was written. This makes very complicated to estimate generic book independent HMM models. In fact, the usual approach is to take part of the book for training and the rest for test. However, ASR HMMs are usually speaker independent.

Two baseline systems have been considered. The first is to transcribe the page using a HTR system. To do this the page must be digitized, the noise must be reduced and the lines segmented. This process is partially manual so it must be considered when evaluating the convenience of using this approach. A 2-gram language model was used in the HTR system. As this system was supposed to run off-line, the WGs were generated without pruning, except that we kept a maximum on 60 incoming edges per node to limit the final size of the WGs.

With this approach we could obtain better WGs that otherwise. However, the computation of such WGs can take several days, what makes it impractical to use in real time decoding. In the second baseline the test transcription is read aloud and the transcriptions come from a dictation ASR system. This system uses the same language model as the HTR system (that is why the perplexities coincide). Nevertheless, in this case the decoder was set up to work in a reasonable amount of time. As a result, the WGs are much smaller. With this conditions, we can expect a better baseline result for HTR than for ASR, since HMM for HTR fit better to test conditions that HMM for ASR, while the language model is the same.

The results are summarized in Table 6.4 and show that the **HTR** system outperforms the **ASR** system. The explanation for this comes naturally from the previous comments. The language model is poorly estimated and the search process depends greatly on the HMM estimates in both cases. Nevertheless, in the HTR case the HMMs have been specifically trained for the particularities of the test (book and writer), whereas the ASR HMMs were trained from a completely different corpus (distinct speakers). All the results were obtained in the same conditions: punctuation marks were not considered; initial, final, and silence/blank symbols were eliminated from the decoder output; all words were transcribed to capital letters.

model	language model	γ	perplexity	WER
HTR	**2GRAM**	—	552	29.2 ± 8.2
ASR	**2GRAM**	0.0	552	43.2 ± 3.3
SHR	**3WG**	1.0	391	45.8 ± 4.0
SHRi	**3WG+2GRAM**	0.2	**54**	$\mathbf{18.6 \pm 2.8}$

Table 6.4: Summary of perplexity and WER for the different approaches to transcription of handwritten historical documents.

An intermediate approach is to use information from both the handwritten text and the speech signal by means of the HTR posterior n-grams of Equation (6.11). This system, referred to as *speech and handwriting recognition* (**SHR**), follows a dictation scenario, as in ASR, but fusing the information from a previous HTR recognition. The parameters needed for this model were estimated using a *leaving-one-out* scheme over the lines of the page. The resulting 3-gram has a perplexity of 391. Although it has quite a better perplexity than the original 2-gram model, this system achieves worse WER results. Nevertheless, confidence intervals at 95% overlap with the ASR system. This high WER is mainly due to the poor smoothing for out-of-vocabulary words when computing the HTR posterior n-grams. HTR lattices contain only a small part of the vocabulary so the rest of the vocabulary was introduced with equal probability (see end of Section 5.4.2). Thus, the probabilities for these words are low and the recognition performance decreases for them.

To prevent from poor estimations of out-of-vocabulary in the HTR posterior 3-grams, this model has been linearly interpolated with the baseline 2-gram as in Equation (6.12), and it is referred to as **SHRi**. Figure 6.4 shows the results when changing the value of the interpolation factor. The ASR baseline is represented by $\gamma = 0$ while the HTR posterior 3-gram system is $\gamma = 1$. The graph shows the WER with confidence intervals at 95% along with the oracle WER. The scale factors where estimated in a *leaving-one-out* scheme over individual utterances. All the interpolated models improve the baseline with 100% *probability of improvement* (POI). It must be noted that almost all set-ups perform in the same range, although when γ approaches 1, the curve slowly raises. However, confidence intervals still overlap among interpolated models. The same behavior can be observed on the *oracle* WER. Best oracle WER achieves an 8.5%, which suggests that there is still room for improvement.

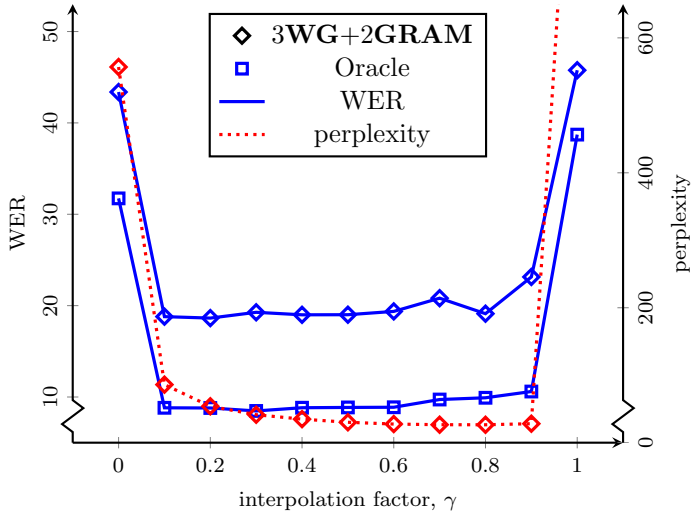


Figure 6.4: WER and oracle WER for dictation systems for different γ values. The first value is an ASR system ($\gamma = 0$), the last a SHR system ($\gamma = 1$). The values within are SHRi systems with γ interpolation factor.

6.3.3 Iterative results

Contrarily to what was assumed in the non-iterative results, here we will assume that HTR WGs cannot be precomputed since we expect close to real time decoding. Thus, all the recognition processes were performed by applying the classical pruning techniques. In consequence, the generated WGs are smaller, although the WER is almost the same. Recognition parameters were tuned on the test set to obtain the optimal results. The results obtained by the iATROS system (including 90% confidence intervals), along with approximated decoding times per sample and feature (considering as feature each element of

the feature vectors), are presented in Table 6.5. These results are similar to those presented in Table 6.4, although HTR results are now obtained applying pruning techniques.

Modality	WER	Time per sample	Time per feature
HTR	29.7 ± 6.7	224 sec	1.72 msec
ASR	43.2 ± 3.3	25 sec	1.56 msec

Table 6.5: Baseline results for handwritten text and speech modalities, along with average decoding times for each sample (in seconds) and each feature (in milliseconds). In contrast with Table 6.4, the **HTR** results were obtained applying pruning techniques.

As we saw in Table 6.4, HTR results are significantly better than speech results. The different magnitude of the confidence intervals is caused by the size of the test set (24 lines for HTR and 120 sentences for ASR). Moreover, decoding times are an order of magnitude lower for ASR than for HTR, given that the HTR feature extraction generates approximately ten times more features. In fact, the time per feature is almost the same for both systems.

The results obtained for the iterative process are presented in Tables 6.6 and 6.7, for HTR and ASR start conditions, respectively. The process stops when the hypothesis of the non-starting modality does not change with respect to the hypothesis in the previous iteration. In any case, the process was limited to 10 iterations. The initial decoding is performed in one of the baseline systems (HTR or ASR) in Table 6.5. Then, the next step is obtaining the 3**WG** model from the recognition WG (according to Equation (6.11)) and perform the interpolation with the original 2-gram (Equation (5.19)). The posterior scale and interpolation parameters were optimized in previous experiments and were kept with the same value along all the process. The decoding parameters for the iATROS system were kept to those used in the baseline experiments.

Iterative from initial HTR

Starting from HTR recognition, a large reduction in WER is obtained by only applying the ASR step on the new language model. The obtained result differs from that obtained in Table 6.4. This is justified by the fact that the initial HTR WGs are less dense since the pruning technique has been applied. In contrast, decoding times are two orders of magnitude lower than that of the WGs not pruned. In any case, here the differences between baseline HTR and ASR are also statistically significant, since confidence intervals do not overlap.

In the iterative process which starts with HTR, the distance between confidence intervals of ASR and the baseline HTR gets higher in each iteration until convergence, although differences are not significant for ASR results from initial iteration to convergence. This similarity between these results is caused

Iteration	Modality		Time per sample
	HTR	ASR	
0	29.7 ± 6.7	20.6 ± 2.4	256 sec.
1	25.8 ± 2.7	20.1 ± 2.4	470 sec.
Convergence	25.4 ± 2.7	**19.9 ± 2.4**	585 sec.

Table 6.6: WER and time results for the iterative process starting from the HTR process. In italics, the baseline result. Convergence is assumed when hypothesis of ASR in one iteration does not change from the hypothesis of the previous iteration.

by the small number of iterations required to obtain convergence: most of the samples (80%) converge with an only iteration (10% require 2 iterations and only 2.5% do not converge within a limit of 10 iterations). Consequently, we can assume that using only one iteration is enough to obtain the good results. This reduces the time needed for obtaining the final hypothesis.

However, in this iterative process starting with HTR, the iterative HTR results do not present a statistically significant improvement with respect to baseline. Moreover, HTR results are always worse than ASR results. In this case, the cause can be the decoding parameters, that were kept to the same value than in the baseline HTR experiment without performing an optimization on the ASR WGs that provided the language model for the iterative HTR decoding.

Error analysis was centered in some special cases that are particular to this HTR task:

- Hyphenated words: including the first part (with an hyphenation symbol - at the end) and the second part (that starts in the following line).
- Abbreviations: in this case, the words '=', 'NTRA', 'S', 'STA', and 'SRA'; they are pronounced as whole words in ASR but kept as abbreviations in HTR.
- Numbers: in this case, '(16)', '38', and 'TREINTA Y CUATRO'; in ASR, the same lexical model represents different words (e.g., '(16)' and '16').

The comparison between the results of convergence HTR and ASR showed small differences, that concentrated mainly on abbreviations '=' and 'STA', and on number '(16)'. ASR presents a quite lower error rate in these cases with respect to those obtained by HTR. This can be caused by the scarce presence of the corresponding symbols ('=', numbers) in the training test for HTR, whereas in ASR the corresponding phones are as usual as those of other words. This causes a poor estimation of the HMM associated to the symbols, which explains the differences in this case and the better results of the ASR variant.

Iteration	Modality		Time per sample
	ASR	HTR	
0	43.2 ± 3.3	25.5 ± 2.5	232 sec.
1	18.4 ± 2.6	24.3 ± 2.5	462 sec.
Convergence	**18.3 ± 2.4**	24.1 ± 2.5	1244 sec.

Table 6.7: WER and time results for the iterative process starting from the ASR process. In italics, the baseline result. Convergence is assumed when hypothesis of HTR in one iteration does not change from the hypothesis of the previous iteration.

Iterative from initial ASR

Starting from ASR recognition (Table 6.7), the only application of HTR using the language model derived from the output WG obtains a dramatical decrease of WER with respect to baseline ASR results. However, when comparing with the HTR baseline results, differences are not significant.

When starting with the iterative process, the ASR results become significantly better than the baseline results for any of the modalities, although HTR results present a worse WER. This behavior is similar to that presented by the iterative process starting from HTR, where in each iteration HTR results are far worse than ASR results. As happened in that previous case, the configuration of the decoding parameters seem the cause of this result.

Anyway, ASR results in this iterative process are the best results that are obtained with the test set (18.3% of WER with respect to the original baseline of 29.7% in the HTR baseline). Moreover, although the number of iterations for convergence is higher that in the iterative process starting from HTR (only a 44% of the samples converge in one iteration and more than a 28% do not present convergence), results in first iteration for ASR are very similar to convergence results. Thus, only one iteration is enough for having an accurate result, which implies a faster process.

Error analysis showed similar conclusions that those of the iterative process starting from HTR. The lack of convergence in the HTR hypothesis during this process was caused mainly by the alternative appearance of two very similar (but different) hypothesis in each step (i.e., hypothesis A appeared in odd iterations and hypothesis B appeared in even iterations).

6.4 Summary of contributions

In this chapter we have described how to improve speech-enabled systems. The explored techniques can be encoded in the form of n-grams so they can be seamlessly integrated in state-of-the-art ASR software. The systems that

do not take information from the context of the target sentences exhibit an increase in the recognition performance with respect to competing alternatives. Furthermore, we have shown that if the integration of all the sources of information is carried out early in the recognition process, the achieved WER improvements are remarkable. In addition, if this integration is performed in the ASR step, the generated WGs improve in quality so that techniques using ASR WGs can benefit from it.

The adoption of the iterative approach in HTR dictation shows a greater improvement in the results with respect to the baseline unimodal systems, although not always significantly better than multimodal non-iterative results. The influence of the initial modality in behavior of the multimodal iterative approach was studied for the two modalities. Results show that starting from ASR allows for a better final performance. In all cases of the iterative process, HTR results present a poorer result that their ASR counterparts, but their decoding parameters seem not as optimal as the decoding parameters of the ASR systems.

Although errors are significantly less than those of baseline systems and time response is appropriate for the transcription task, more work can be done to improve these two aspects and obtain a better performance in a real system. On the one hand, HTR decoding needs a better tuning for the recognition parameters, that can be obtained in a similar way to those obtained for the ASR decoding; the improvement in the HTR decoding could cause a positive impact in the final performance of the iterative systems, both in terms of WER and convergence time. On the other hand, an integrated decoding of HTR and ASR is a suggestive alternative to the iterative paradigm that can reduce dramatically the final time and obtain better quality in the automatic transcriptions.

However, as results are still far from perfect, the error rate may prevent the users from effectively using the speech modality. Thus, we should make an effort to reduce the WER to a point where speech-enabled interfaces are usable. We could take advantage that the user is in front of the screen to capture face expressions and lip movement to perform audio-visual speech recognition. Finally, the multimodal aspects of this problem should be studied, i.e. how the speech interface and keyboard and mouse can be used alternatively or simultaneously. Eventually, human evaluation should be carried out to assess the proposed scenarios in a real-like situation.

The speech-enabled IMT experiments were published in:

- **V. Alabau**, L. Rodríguez-Ruiz, A. Sanchis, P. Martínez-Gómez, and F. Casacuberta. On multimodal interactive machine translation using speech recognition. In *Proc. of the International Conference on Multimodal Interaction (ICMI'11)*, p. 129–136, 2011.

On the other hand, the *sight transcription* system for historical handwritten documents lead to the following publications:

- **V. Alabau**, C. D. Martínez-Hinarejos, V. Romero, and A. L. Lagarda. An iterative multimodal framework for the transcription of handwritten historical documents. *Pattern Recognition Letters*, in press:–, 2012.
- **V. Alabau**, V. Romero, A. L. Lagarda, and C. D. Martínez-Hinarejos. A Multimodal Approach to Dictation of Handwritten Historical Documents. In *Proc. of Interspeech'11*, p. 2245–2248, 2011.

Bibliography

S. Bengio. Multimodal speech processing using asynchronous Hidden Markov Models. *Information Fusion*, 5(2):81 – 89, 2004.

P. Brown, S. Chen, S. D. Pietra, V. D. Pietra, S. Kehler, and R. Mercer. Automatic speech recognition in machine aided translation. *Computer Speech and Language*, 8:177–187, 1994.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The Mathematics of Machine Translation. *Computational Linguistics*, 19(2):263–311, 1993.

F. Casacuberta, E. Vidal, and D. Picó. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38:1431–1443, 2005.

B. Dragsted, I. M. Mees, and I. G. Hansen. Speaking Your Translation: Students' First Encounter with Speech Recognition Technology. *The Translation & Interpreting*, 3(1): 10–43, 2011.

M. Dymetman, J. Brousseau, G. Foster, P. Isabelle, Y. Normandin, and P. Plamondon. Towards an Automatic Dictation System for Translators: the TransTalk Project. In *Proc. of International Conference on Spoken Language Processing (ICSLP'94)*, p. 193–196, 1994.

F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

S. Khadivi and H. Ney. Integration of Speech Recognition and Machine Translation in Computer-Assisted Translation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1551–1564, 2008.

M. LaLomia. User acceptance of handwritten recognition accuracy. In *Proc. of the Conference on Human Factors in Computing Systems (CHI'94)*, p. 107–108, 1994.

P. Liu and F. Soong. Graph-Based Partial Hypothesis Fusion for Pen-Aided Speech Input. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):478–485, 2009.

M. Luján-Mares, V. Tamarit, V. Alabau, C. D. Martínez-Hinarejos, M. Pastor-i Gadea, A. Sanchis, and A. H. Toselli. iATROS: A speech and handwritting recognition system. In *V Jornadas en Tecnologías del Habla (VJTH'2008)*, p. 75–78, 2008.

J. Mariño, R. Banchs, J. Crego, A. de Gispert, P. Lambert, J. Fonollosa, and M. Costa-jussà. N-gram-based Machine Translation. *Computational Linguistics*, 32 (4):527–549, 2006.

F. J. Och and H. Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, p. 295–302, 2002.

A. Ogawa, K. Takeda, and F. Itakura. Balancing acoustic and linguistic probabilites. In *Proc. of the IEEE Conference Acoustics, Speech, and Signal Processing (ICASSP'98)*, p. 181–184, 1998.

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. Thot: a Toolkit To Train Phrase-based Statistical Translation Models. In *Proc. of the MT Summit X*, p. 141–148. 2005.

S. Oviatt and R. VanGent. Error resolution during multimodal human-computer interaction. In *Proc. of the Fourth International Conference on Spoken Language (ICSLP'96)*, volume 1, p. 204–207, 1996.

M. Paulik, C. Fügen, S. Stüker, T. Schultz, T. Schaaf, and A. Waibel. Document driven machine translation enhanced ASR. In *Proc. of the 9th European Conference on Speech Communication and Technology (Eurospeech'05)*, p. 2261–2264, 2005.

A. Reddy and R. Rose. Integration of Statistical Models for Dictation of Document Translations in a Machine-Aided Human Translation Task. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2015–2027, 2010.

L. Rodríguez-Ruiz. *Interactive Pattern Recognition Applied to Natural Language Processing*. Tesis doctoral en informática, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2010.

M. Shilman, D. S. Tan, and P. Simard. CueTIP: a mixed-initiative interface for correcting handwriting errors. In *Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST'06)*, p. 323–332, 2006.

E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-Assisted Translation Using Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 14(3):941–951, 2006.

Chapter 7

Conclusions

Chapter Outline

Structured prediction systems are becoming pervasive. For instance, almost any modern mobile device allows to use speech or handwriting to introduce text; machine translation is a hot topic in translation agencies; OCR has a long software tradition; and the transcription of historical documents raises a high interest among historians. Usually, these technologies are not able to produce results that match high-quality standards. Hence, it is necessary a human operator to amend the automatic output to obtain the desired result.

The work presented in this thesis has focused on how this interaction between humans and automatic systems can be carried out. On the one hand, we have studied how the system and user can collaborate efficiently. On the other hand, we have delved into which modalities can be used for this collaboration to happen more comfortably. The following sections are a summary of the main contributions proposed in this thesis, future work and scientific publications produced as a result of this work.

7.1 Summary

One of the better-known interaction protocols for structured prediction problems is a sequential interactive protocol [Vidal et al., 2007]. In Chapter 3, we questioned the decision rule that was being used in the state-of-the-art and found an algorithm that produces an optimum decision. Additionally, we established a relationship between the optimum algorithm and the standard in which the latter is a maximum approximation to the former. In practice, both decision rules showed similar results with respect to user effort. However, the optimum decision rule outperformed significantly the standard in some of the tasks.

Left-to-right interaction forces the user to supervise the whole set of outputs. In this regard, in Chapter 4 we studied techniques to avoid this problem by allowing the system to propose what outputs the user should amend. We proved that the optimum decision rule consists in correcting first the structures that the systems expects to have more errors. This strategy was compared to other strategies borrowed from active learning. We found that the optimum decision rule performed better than the rest whenever the exact algorithm for computing the expected error could be used. In contrast, when using an approximated algorithm, sorting by entropy was the best strategy. Moreover, sorting by posterior probability was almost as good as sorting by entropy, with the advantage that its computation is usually much easier. Additionally, we observed similar results when these strategies were used in a streaming scenario, for which the optimum strategy is to establishing a threshold.

On the other hand, in the second part of Chapter 4, we studied an active protocol for correcting elements of a given structure. For the basic strategy, which is based on posterior probabilities at element level, the posterior probability

can be maximized or minimized. When aiming at reducing the number of corrections, maximizing the posterior probability obtained the best results. This fact was consistent with the findings in the left-to-right protocol, since in that case the optimum solution was also to maximize the posterior probability of the element. This opens an avenue to find an optimum decision rule for active interaction at element level. However, minimizing the posterior probability obtained the worst results. Intuitively, we can draw from this that minimizing the posterior probability may be useful for minimizing supervision effort, as in structure level interaction, but not for minimizing the number of corrections. What is more, we could say that both goals are contradictory.

In the second part of the thesis, several scenarios for multimodal interaction were studied. To begin with, e-pen was devised as an alternative input modality to keyboard and mouse for interactive machine translation (cf. Chapter 5). We developed several strategies to provide a more robust handwriting recognition by leveraging contextual information. Two main approaches were proposed based on word-based and phrase-based translation models, respectively. The experimentation results suggested that using the more reliable and less context-aware word-based models resulted in better improvements than using phrase-based models. In the end, word-based models were interpolated with n-gram prefix models to give more information regarding the context in the target side, resulting in additional improvements. Next, we studied a mechanism for recovering from recognition errors by using n-best lists. Encouragingly, within only the best 5 candidates, the error was reduced to a quarter. Finally, a study of pen gestures revealed that IMT systems can be built to react to these gestures in a way that the overall user effort can be further reduced.

Another modality that is natural for humans and suitable for interactive systems is speech. In this case, we considered two scenarios in Chapter 6. The first one was an IMT system. The user was supposed to use a pointer to indicate where the corrections should be inserted. Then the user should utter a correction that could be composed of more than one words. We explored new techniques for fusing the translation and speech inputs to provide a more reliable recognizer. The techniques that we used were similar to those used with e-pen interaction, obtaining noticeable results. However, by contrast with e-pen interaction, in this case the results showed that by using information from the phrase-based models the recognition accuracy improved with respect to word-based models. We hypothesize that this is due to the fact that HTR morphologic models are more accurate than the ASR ones and, thus, contextual information of the target side is more critical. The second scenario considered for speech interaction was sight transcription. There, the user was supposed to dictate full sentences of a historical document. The output word graph of the HTR system was used to generate a language model for the ASR system, and the other way around. Empirical results showed remarkable improvements with respect to the baseline system in any of each directions. Additionally, we could observe that using the most accurate system to generate the language

model for the worse system is significantly better than the contrary. Additionally, an iterative decoding algorithm was proposed which was able to achieve additional gains.

Finally, we presented an assessment of an IMT prototype with real users in Appendix A. Two rounds of evaluations were performed comparing the IMT prototype with a PE prototype (the IMT prototype but with disabled interaction capabilities). The conclusions we could draw from the first round was that the design of user interface, from a technical point of view, is critical for the users to accept new technologies. In the second round we solved some of the issues arisen. The results were encouraging, showing that users where more favorable to use the interactive prototype than the PE prototype. In addition, we could infer some lines of improvement for future research in interactive systems. First, a system should propose new hypotheses only if it is sure that the new suggestions are significantly better than the previous ones. Secondly, suggestions should not be changed too often since, arguably, it imposes more cognitive load on the user. Note that this second argument can result from the first one, i.e., if the system is not sure of what to suggest, it will change its mind often. Lastly, and most importantly, a system should not change elements that the user has already validated, which reduces the user's trust in the system.

7.2 Future work

It is quite common that the research process opens more questions that it closes. The case of this thesis is not different. During the development of the contents presented here, several avenues for improvement have arisen.

First, the optimum algorithm for sequential interaction may provide a new way of understanding search-based structured prediction [Daumé III et al., 2009]. To this regard, the word graphs of the optimum algorithm generated by Proposition 3.3 could be used as a constrained search space where search-based structured prediction can operate. This would allow to introduce arbitrary features in the search process that would not be possible otherwise, e.g., gender and number agreement. In addition, the cost function we used in sequential interaction is a rather simplistic since it only takes into account characters or words being substituted, as an approximation to the actual productivity. This work should be extended with a more realistic user modeling, in the line of [Foster et al., 2002], where not only the mechanical effort is considered but also the cognitive effort.

Second, element level active interaction has left some questions unanswered. In the first place, the optimum algorithm is still to be unveiled. A starting point is the *most confident* strategy since, intuitively, it works with the same principles than the optimum algorithm for the sequential problem: let the user supervise an highly confident element since supervision has zero cost; then expect that corrective feedback is able to correct low confident elements. Nonetheless, it is

safe to assume that supervision is cheap, whereas correction expensive? This assumption seems reasonable for sequential interaction since the user needs to supervise the whole output to find the errors, anyway. However, in active interaction it should not be compulsory to supervise all the output elements. Unfortunately, from the experiments in Section 4.4.2 we can deduce that minimizing supervision and minimizing corrections are contradictory goals. Thus, a different cost function should be defined, for instance, assigning cost one to supervision and cost two to correction. This should change how the different strategies perform. Probably, as suggested by Culotta et al. [2006], a new possible strategy would be to amend elements with high confidence which are close to elements with low confidence. At last, an active interaction protocol should be devised for problems where the size of the output is not known.

Concerning multi-modal fusion, some experiments have been left for future work. On the one hand, word-based model M2 was not evaluated when the user could introduce more than one word, though it achieved the best performance in the experiments with isolated words. That was because we could not codify M2 probabilities in the form of n-grams. Hopefully, our corpus showed that the average length of the interactions was between 2 and 3 words. This suggests that M2 probabilities could be computed explicitly for combinations of words, for instance, up to 3 words, which should be computationally affordable. Furthermore, it is still unclear how the size of the word graphs affects the quality of the fusion. Extensive experiments should be carried out to find a compromise between the cost of generating a dense graph and the decoding accuracy. On the other hand, other multimodal fusion challenges have appeared recently [Pastor-i-Gadea et al., 2010][1]. That paper presents the biMod-IAM-PRHLT-2 corpus, a bimodal dataset of on-line and off-line handwritten text. Previous attempts to integrate these signals have shown that a simple linear interpolation approach is hard to beat [Pastor-i-Gadea and Paredes, 2010]. Although the techniques developed in this thesis do not seem good approaches for the biMod-IAM-PRHLT-2 corpus at a first sight, the experience gained may provide insights to design specific algorithms that could be useful.

Finally, one of the goals of applied research is that research results can be transferred to real technology. During the evaluation of the prototypes in Appendix A, we observed that users were very sensitive to the usability aspects of the UI, which can in turn hinder the benefits of the techniques being evaluated. Also, we noticed that the new features implemented were decided on the basis of researchers' interests and not on what may have more impact in the way user interact with the tools. Both facts contribute to create systems that, when evaluated in a between-subjects design, do not show significant improvements with respect to the control condition. What is more, in some cases the new technique performs worse as a result of a bad interaction experience introduced by the complexities of the new technique. Therefore, we must reconsider how we tackle the human evaluation of the prototypes. One possible approach

[1] https://prhlt.iti.upv.es/page/contests/bimodal2/index.php

would be to follow a *lean* research process, as in *lean manufacturing* [Womack and Jones, 2003], *lean software development* [Poppendieck and Poppendieck, 2003] or *lean startup* [Ries, 2011], in an attempt to find what are the actual problems to be solved. For that, it would be ideal to start from a working product with standard quality and features, and with a significant user base. Then, we could perform A/B (split) testing [Kohavi et al., 2009] to identify which problems to research on.

7.3 Scientific publications

The content of this thesis has lead to publications in international workshops, conferences and journals. Each chapter ends with a relation of publications resulting from it. Besides, in this section we review these publications but listed now by their relevance.

To begin with, an article has been submitted to a journal with estimated impact factor in year 2012 of 2.632:

- **V. Alabau**, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition in Interactive Machine Translation. *Pattern Recognition*, submitted to(–), 2013.

The article is in the second round of the review process with a favorable review in the first round.

Second, two articles have been published in a journal with estimated factor of 1.266 in year 2012:

- **V. Alabau**, C. D. Martínez-Hinarejos, V. Romero, and A. L. Lagarda. An iterative multimodal framework for the transcription of handwritten historical documents. *Pattern Recognition Letters*, in press:–, 2012.
- **V. Alabau**, A. Sanchis, and F. Casacuberta. On the Optimal Decision Rule for Sequential Interactive Structured Prediction. *Pattern Recognition Letters*, 33(6):2226–2231, 2012.

Moreover, two research papers have been published in international conferences ranked A by the Computing Research and Education Association of Australasia (CORE):

- **V. Alabau**, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*, p. 389–394, 2011

- **V. Alabau**, V. Romero, A. L. Lagarda, and C. D. Martínez-Hinarejos. A Multimodal Approach to Dictation of Handwritten Historical Documents. In *Proc. of Interspeech'11*, p. 2245–2248, 2011.

There have been also some publications indexed as CORE B:

- **V. Alabau**, L. A. Leiva, D. Ortiz-Martínez, and F. Casacuberta. User Evaluation of Interactive Machine Translation Systems. In *Proc. of the 16th Annual Conference of the European Association for Machine Translation (EAMT'12)*, p. 20–23, 2012.
- **V. Alabau**, L. Rodríguez-Ruiz, A. Sanchis, P. Martínez-Gómez, and F. Casacuberta. On multimodal interactive machine translation using speech recognition. In *Proc. of the International Conference on Multimodal Interaction (ICMI'11)*, p. 129–136, 2011.
- **V. Alabau**, D. Ortiz-Martínez, A. Sanchis, and F. Casacuberta. Multimodal Interactive Machine Translation. In *Proc. of the International Conference on Multimodal Interfaces (ICMI-MLMI'10)*, p. 46:1–46:4, 2010.

On the other hand, the following paper has been published in an non-indexed workshop:

- **V. Alabau** and F. Casacuberta. Study of Electronic Pen Commands for Interactive-Predictive Machine Translation. In *International Workshop on Expertise in Translation and Post-editing Research and Application*, 2012.

Additionally, it should be mentioned that the work in active interaction in Chapter 4 is in preparation for publication.

Finally, during the period of this thesis, further work has been carried out that is related to the contents presented here. However, these publications have not been addressed directly in this thesis. It is specially worth of mention the publications derived from the design and implementation of the prototypes where the author of this thesis has played an active role, leading in many cases the design of the architecture and implementation of the prototype. The publications derived from the prototypes are:

- **V. Alabau**, R. Bonk, C. Buck, M. Carl, F. Casacuberta, M. Garcia-Martinez, P. Koehn, L. A. Leiva, B. Mesa-Lao, H. Saint-Amand, C. Tsou-kala, G. Sanchis-Trilles, D. Ortiz-Martínez, and J. González-Rubio. Advanced Computer Aided Translation with a Web-Based Workbench. In *Proc. of the MT SUMMIT XIV (MT-SUMMIT'2013)*, 2013. (CORE B)

- V. Romero, L. A. Leiva, and **V. Alabau**. *Multimodal Interactive Handwritten Text Transcription*, chap. A Web-based Demonstrator of Interactive Multimodal Transcription. 2012.

- D. Ortiz-Martínez, L. A. Leiva, **V. Alabau**, I. García-Varea, and F. Casacuberta. An Interactive Machine Translation System with Online Learning. In *Proc. of the ACL-HLT: System Demonstrations (ACL'11)*, p. 68–73, 2011. (CORE A)

- L. A. Leiva, **V. Alabau**, V. Romero, F. M. Segarra, R. Sánchez-Sáez, D. Ortiz-Martínez, L. Rodríguez-Ruiz, and N. Serrano. *Multimodal Interactive Pattern Recognition and Applications*, chap. Prototypes and Demonstrators. 1st edition edition, 2011.

- D. Ortiz-Martínez, L. A. Leiva, **V. Alabau**, and F. Casacuberta. Interactive Machine Translation using a Web-based Architecture. In *Proc. of the International Conference on Intelligent User Interfaces (IUI'10)*, p. 423–425, 2010. (CORE A)

- **V. Alabau**, J.-M. Benedí, F. Casacuberta, L. A. Leiva, D. Ortiz-Martínez, V. Romero, J.-A. Sánchez, R. Sánchez-Sáez, A. H. Toselli, and E. Vidal. CAT-API Framework Prototypes. In *Proc. of Database and Expert Systems Applications (DEXA), 2010 Workshop on Interactive Multimodal Pattern Recognition in Embedded Systems (IMPRESS 2010)*, p. 264–265, 2010.

- **V. Alabau**, F. Casacuberta, L. A. Leiva, D. Ortiz-Martínez, and G. Sanchis-Trilles. Sistema web para la traducción automática interactiva. In *Actas del XI Congreso Internacional de Interacción Persona Ordenador. INTERACCIÓN2010*, p. 47 – 56, 2010.

- V. Romero, L. A. Leiva, **V. Alabau**, A. H. Toselli, and E. Vidal. A Web-based Demo to Interactive Multimodal Transcription of Historic Text Images. In *Proc. of the 13th European Conference on Digital Libraries (ECDL'09)*, p. 459–460. 2009. (CORE A)

- **V. Alabau**, D. Ortiz-Martínez, V. Romero, and J. Ocampo. A multimodal predictive-interactive application for computer assisted transcription and translation. In *Proc. of the International Conference on Multimodal Interfaces (ICMI-MLMI'09)*, p. 227–228, 2009. (CORE B)

- M. Luján-Mares, V. Tamarit, **V. Alabau**, C. D. Martínez-Hinarejos, M. P. i Gadea, A. Sanchis, and A. H. Toselli. iATROS: A speech and handwritting recognition system. In *V Jornadas en Tecnologías del Habla (VJTH'2008)*, p. 75–78, 2008.

The prototypes have also received the following awards:

- **Accesit (ex aequo)** in the Valencia Idea 2010 competition, funded by the Valencia city council in the category of Information and Communications Technologies. *Nuevas Tecnologías Interactivo-Predictivas Multimodales para el Procesamiento de Lenguaje Natural sobre Internet.* L.A. Leiva, D. Ortiz-Martinez, E.M. Cubel, G. Sanchís-Trilles, V. Romero, R. Sánchez-Sáez, **V. Alabau**, A.H. Toselli, J.A. Sánchez, J.M. Benedí, E. Vidal, and F. Casacuberta

- **Best Demonstration Award** in the 13th European Conference on Digital Libraries, 2009. *A Web-based Demo to Interactive Multimodal Transcription of Historic Text Images.* V. Romero, L. A. Leiva, **V. Alabau**, A. H. Toselli, and E. Vidal.

Lastly, the following is a list of the remaining publications that relate to structure prediction, multi-modality, interactivity, and user studies, but were not explicitly developed for the present thesis:

- L. A. Leiva, **V. Alabau**, and E. Vidal. Error-proof, High-performance, and Context-aware Gestures for Interactive Text Edition. In *Proc. of the Annual Conference Extended Abstracts on Human Factors in Computing Systems (CHI'13 EA)*, p. 1227–1232, 2013. (CORE A)

- L. A. Leiva and **V. Alabau**. An Automatically Generated Interlanguage Tailored to Speakers of Minority but Culturally Influenced Languages. In *Proc. of the Annual Conference on Human Factors in Computing Systems (CHI'12)*, p. 31–34, 2012. (CORE A)

- **V. Alabau** and L. A. Leiva. Transcribing Handwritten Text Images with a Word Soup Game. In *Proc. of the Annual Conference Extended Abstracts on Human Factors in Computing Systems (CHI'12 EA)*, p. 2273–2278, 2012. (CORE A)

- **V. Alabau**, G. Sanchis-Trilles, and L. Rodríguez-Ruiz. *Multimodal Interactive Pattern Recognition and Applications*, chap. Multi-modality for Interactive Machine Translation. 1st edition edition, 2011.

- G. Gascó, **V. Alabau**, J. Andrés-Ferrer, J. González-Rubio, M.-A. Rocha, G. Sanchis-Trilles, F. Casacuberta, J. González, and J.-A. Sánchez. ITI-UPV system description for IWSLT 2010. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT'10)*, 2010.

- M. Luján-Mares, C. D. Martínez-Hinarejos, **V. Alabau**, and A. Sanchis. Some issues on the Expectation-Maximisation process for Maximum Likelihood Linear Regression. In *VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop (FALA 2010)*, 2010.

- A. L. Lagarda, **V. Alabau**, F. Casacuberta, R. Silva, and E. D. de Liaño. Statistical Post-Editing of a Rule-Based Machine Translation System. In *Proc. of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL'09)*, p. 217–220, 2009. (CORE A)

- M. Luján-Mares, C. D. Martínez-Hinarejos, and **V. Alabau**. A Study on Bilingual Speech Recognition Involving a Minority Language. In Z. Vetulani and H. Uszkoreit, editors, *Human Language Technology. Challenges of the Information Society, Third Language and Technology Conference, LTC 2007, Poznan, Poland, October 5-7, 2007, Revised Selected Papers*, p. 36–49. 2009.

- L. Tarazón, D. Pérez, N. Serrano, **V. Alabau**, O. Ramos-Terrades, A. Sanchis, and A. Juan. Confidence Measures for Error Correction in Interactive Transcription of Handwritten Text. In *Proc. of the 15th International Conference on Image Analysis and Processing (ICIAP'09)*, p. 567–574, 2009. (CORE B)

- A. L. Lagarda, **V. Alabau**, C. D. Martínez-Hinarejos, A. H. Toselli, V. Romero, J. R. Navarro-Cerdan, and E. Vidal. Computer-Assisted Handwritten Text Transcription Using Speech Recognition. In *V Jornadas en Tecnología del Habla (VJTH'2008)*, p. 229–232, 2008.

- M. Luján-Mares, C. D. Martínez-Hinarejos, and **V. Alabau**. Evaluation of several Maximum Likelihood Linear Regression Variants for Language Adaptation. In *Proc. of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008. (CORE C)

- M. Luján-Mares, V. Tamarit, R. Paredes, **V. Alabau**, and C. D. Martínez-Hinarejos. El sistema de identificación de la lengua de PRHLT. In *V Jornadas en Tecnologías del Habla (VJTH'2008)*, p. 110–111, 2008.

- M. Luján-Mares, C. D. Martínez-Hinarejos, and **V. Alabau**. A study on bilingual speech recognition involving a minority language. In *Proc. of the 33rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, p. 138–142, 2007.

- V. Romero, **V. Alabau**, and J.-M. Benedí. Combination of N-grams and Stochastic Context-Free Grammars in an Offline Handwritten Recognition System. In *Proc. of the 3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'07*, p. 467–474. 2007. (CORE C)

- **V. Alabau**, F. Casacuberta, E. Vidal, and A. Juan. Inference of Stochastic Finite-State Transducers Using N-gram Mixtures. In *Proc. of the 3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'07*, p. 282–289. 2007. (CORE C)

- **V. Alabau**, A. Sanchis, and F. Casacuberta. Improving Speech-to-Speech Translation Using Word Posterior Probabilities. In *Proc. of the MT SUMMIT XI (MT-SUMMIT'07)*, 2007. (CORE B)

- **V. Alabau**, A. Sanchis, and F. Casacuberta. Using Posterior Probabilities in Lattice Translation. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT'07)*, p. 131–136, 2007.

- **V. Alabau** and C. D. Martínez-Hinarejos. Bilingual speech corpus in two phonetically similar languages. In *Proc. of the International Language Resources and Evaluation (LREC'06)*, p. 1624–1627, 2006. (CORE C)

- **V. Alabau** and C. D. Martínez-Hinarejos. Bilingual Speech Recognition in Two Phonetically Similar Languages. In *IV Jornadas en Tecnologia del Habla*, p. 197–202, 2006.

- **V. Alabau**, J. Andrés-Ferrer, F. Casacuberta, J. Civera, J. García-Hernández, A. Giménez-Pastor, A. Juan, A. Sanchis, and E. Vidal. The naive Bayes model, generalisations and applications. In F. Plá, P. Radeva, and J. Vitrià, editors, *Pattern Recognition: Progress, Directions and Applications*, p. 162–179. 2006.

- **V. Alabau**, J.-M. Benedí, F. Casacuberta, A. Juan, C. D. Martínez-Hinarejos, M. P. i Gadea, L. Rodríguez-Ruiz, J.-A. Sánchez, A. Sanchis, and E. Vidal. Pattern Recognition Approaches for Speech Recognition Applications, 2006.

- J. García-Hernández, **V. Alabau**, A. Juan, and E. Vidal. Bernoulli mixture-based classification. In A. R. Figueiras-Vidal, editor, *Proc. of the LEARNING04*, 2004.

Bibliography

A. Culotta, T. Kristjansson, A. McCallum, and P. Viola. Corrective Feedback and Persistent Learning for Information Extraction. *Artificial Intelligence*, 170:1101–1122, 2006.

H. Daumé III, J. Langford, and D. Marcu. Search-based Structured Prediction. 2009.

G. Foster, P. Langlais, and G. Lapalme. User-friendly Text Prediction for Translators. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*, p. 148–155, 2002.

R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1): 140–181, 2009.

M. Pastor-i-Gadea and R. Paredes. Bi-modal Handwritten Text Recognition ICPR'10 Contest report. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR'10)*, 2010.

M. Pastor-i-Gadea, A. Toselli, F. Casacuberta, and E. Vidal. A Bi-modal Handwritten Text Corpus: Baseline Results. In *Proc. of 20th International Conference on Pattern Recognition (ICPR'10)*, p. 1933–1936, 2010.

M. Poppendieck and T. Poppendieck. *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional, 2003.

E. Ries. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. 2011.

E. Vidal, L. Rodríguez, F. Casacuberta, and I. García-Varea. Interactive Pattern Recognition. In *Proc. of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI'07)*, volume 4892, p. 60–71. 2007.

J. P. Womack and D. T. Jones. *Lean Thinking*. Free Press, 2003.

Appendix A

Human Evaluation of an Interactive Machine Translation Prototype

Chapter Outline

A.1 Introduction

Following the SISP paradigm, recent developments in search algorithms and software architecture have allowed multi-user web-based prototypes. These systems have grown in features, allowing e.g. advanced multimodal interaction or confidence estimations on the predicted output. However, these features also add extra complexity to the prototypes, making imperative to test their effectiveness with respect to technology dissemination. While pure data-driven evaluations have already shown that SISP is a promising technology [Barrachina et al., 2009; Romero et al., 2011], surprisingly, formal human evaluations are highly scarce in the literature.

Langlais et al. [2002] performed a human evaluation on their IMT prototype. They emulated a realistic working environment in which the users could obtain automatic completions for what they were typing. Users reported an improvement in performance; however, raw productivity decreased by 17%, although the users appreciated the tool and were confident to improve their productivity after proper training. That work was extended in the TT2 project [Casacuberta et al., 2009], where the performance tended to increase as the participants, over a 18-month period, grew accustomed to the system. A slightly different approach was studied in [Koehn, 2010]. There, monolingual users evaluated a translation interface supporting IMT predictions and the so-called 'translation options'. When translating from undecipherable languages (as Chinese or Arabic for an English speaker), richer assistance improved user performance.

In this chapter we do not use any of the optimal decision rules or multimodal interfaces we have developed during this thesis. Instead, we focus more on the technology challenges that arise during the development of SISP software, and how human factors contribute to the success of SISP systems. We describe our experiences developing and evaluating two IMT prototypes with real users. On the one hand, an initial, full-featured advanced version that resulted from the developments of the *Multimodal Interaction in Pattern Recognition and Computer Vision* (MIPRCV) project[1] which turned out to be cumbersome and more sensitive to programming errors. On the other hand, a simplified version of the original prototype that focused on the SISP auto-completion capabilities that was favorably received. Our results identify important design issues, which open a discussion regarding how SISP systems should be deployed.

A.2 Learning from previous experiences

In our research group, other SISP prototypes had been developed. For instance, Lagarda et al. [2003] developed a desktop application for IMT and Romero et al. [2009] deployed a web application for IHT. However, each of them presented its own problems that should be addressed. First, regarding the IMT demo

[1] http://miprcv.iti.upv.es

in [Lagarda et al., 2003], the interface was tightly coupled with the IMT engine. This made the migration to new versions of the IMT engine difficult. In addition, the design did not allow to easily add new features, as e-pen or speech interaction, in addition to keyboard and mouse. Moreover, for a potential user to try and test the prototype, the application was to be downloaded and installed, which only worked under Linux, and the models, which could occupy an extensive amount of space. Second, the demo in [Romero et al., 2009] was a web based demo available in Internet. This was good for dissemination purposes since new users could have the hands on the prototype quite quickly. On the contrary, it was composed of a difficult-to-debug set of shell scripts and multiple users could not access the prototype at the same time. Finally, and most importantly, all those prototypes did not allow an easy interchange of the SISP engine. Thus, different strategies for producing the predictions and performing the multimodality could not be tested easily.

A crucial aspect of the prototypes, then, should be to serve as a test bed for new interaction modalities such as speech, or to experiment with other NLP problems, e.g., speech transcription or text prediction. In addition, it would also interesting to design of different user interfaces (or views) to approach same NLP problem in a more ergonomic and comfortable way.

A.3 Defining a multimodal interactive prototype

Within the MIPRCV project, several multimodal interactive tools for natural language processing were being developed. Hence, since all the interactive systems were approached with the sequential ISP protocol (Section 3.2), it seemed logical to design a prototype that could fit all of them, where the back-end systems and interfaces could be easily interchanged. The purpose of the prototypes was not to have a professional finished product but to serve as a showcase for the *multimodal interactive* (MI) theory and tools developed within the MIPRCV project.

A.3.1 Objectives

Therefore, we established the objectives of the prototypes, which can be summarized in the following items:

- Develop a fully functional prototype of a SISP system. Several prototype interfaces can be created, if necessary, to test and study the usability of the system.
- Make the prototype appealing and easily accessible to the general public. In a few clicks, the user should be able to start testing the system, the interface should be intuitive and help should be at hand. Furthermore, there should be variety in language pairs and corpora

- Analyze the strengths and weaknesses of the SISP approach in a real-like scenario. This analysis should lead to the discovery of new ways of interaction.

A.3.2 Functional requirements

Based on the research carried out within the MIPRCV project, the following list of the features was identified to be desirable for a SISP prototype:

Suffix prediction. When the user corrects the proposed solution, a new, hopefully improved suffix should be proposed.

User actions. The interface should allow the user to perform the following actions on the proposed suffix:

Substitute. Substitute the first word or character of the suffix.

Delete. Delete the first word or character of the suffix.

Insert. Insert a word before the suffix.

Reject. The rejected word must not to appear in the following proposals.

Accept. Validate a whole output, i.e., acknowledge the system that the output is correct.

Keyboard shortcuts. The user should be able to perform certain actions by means of keyboard shortcuts.

Mouse gestures. The user should be able to perform certain actions by means of mouse gestures.

Pen interactivity. The user should be able to correct the words by handwriting with a digital pen or tablet.

Word and character interaction. : Word level and character level operations should be allowed.

Confidence measures. Confidence measures should be shown to indicate which words are considered to be correct and which ones to be incorrect.

On-line adaptation. The system should be able to learn from validated translations to help improve the quality of future translations.

Document visualization. At any time, the user should be able to visualize the original document, as well as draft of the current decoding in the proper formating.

Document selection. A list of documents should be presented to the user so that she can test the prototype under different conditions, i.e. several corpora and language pairs.

Prediction disabling. The user should be able to disable the predictive system and back off to a post-editing system.

Activity logging. A logging system should keep track of every detail of the user interaction for a later analysis of the results and interaction replay.

A.4 Architecture Design

As we have mentioned, the two main aspects for the prototype architecture are accessibility and flexibility. The former is necessary to reach a larger number of potential users. The latter allows the researches to test different techniques and interaction protocols reducing the implementation effort. For that reason, an *application programming interface* (API) [Alabau et al., 2010a] was developed. Based on the MI protocol, we extracted a generic subset of primitives for most common NLP tasks, and designed a client-server API and library that allows client and server applications to communicate through sockets. Three basic functions summarize the API:

set_source : selects the input to be transcribed or translated.

set_prefix : sets the longest error free prefix and the amendment of the first error as a character string.

set_prefix_online : sets the longest error free prefix and the amendment of the first error with pen strokes.

This API allows a neat separation between the client interface and the actual SISP system by exposing a well-defined set of functions extracted from the multimodal SISP protocols, and by using a network communication protocol to link desktop and web interfaces with SISP back-ends. A diagram of the architecture is shown in Figure A.1.

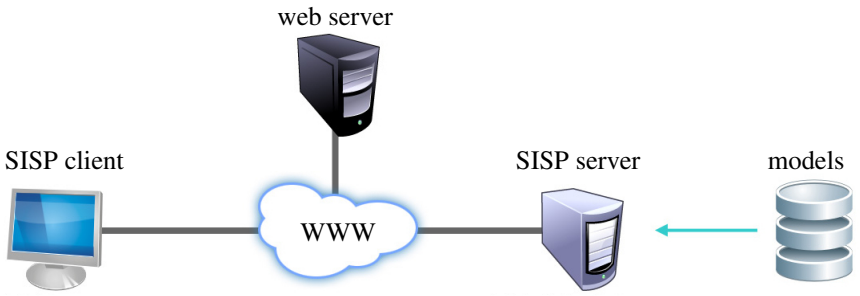


Figure A.1: Illustration of the MIPRCV client-server architecture for SISP problems. The web server is a regular web server that provides an HTML user interface to the SISP client, which in this case is a regular web browser. Moreover, the SISP server deals with the suffix prediction and other decoding algorithms that nurture from statistical models.

Two client interfaces were developed using the MIPRCV API: an installable application for IMT and IHT [Alabau et al., 2009] (Figure A.2), and web-based demos for IMT, IHT and IPP [Alabau et al., 2010b; Ortiz et al., 2010; Sánchez-Sáez et al., 2010] (Figure A.3). Furthermore, several backends were also deployed. The backend systems were easily interchanged by just selecting

a different host name and port. On the web-based demo, the client-server communication is made asynchronously via Ajax, providing thus a richer interactive experience. The interface was built using HTML, Javascript and Actionscript. All corrections are stored in plain text logs on the server, so the user can retake them in any moment, also allowing other users to help to translate the full document(s). In the case of the installable application, the interface uses C as programming language and GTK as graphical toolkit.

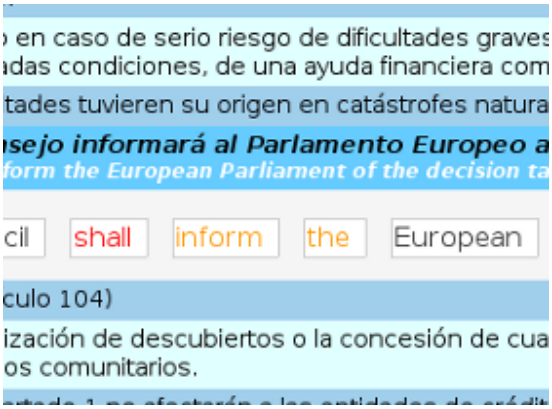


Figure A.2: Detail of the web-based interface.

Finally, advanced multimodal capabilities were added to the web based demo. First, a handwritten text recognition system [Alabau et al., 2010c, 2011], based on the iAtros system [Luján-Mares et al., 2008] was added to perform e-pen interaction. Mouse click operations [Sanchis-Trilles et al., 2008a,b] and, insertion and deletion operations were also implemented. Furthermore, the prototype can show confidence measures at word level [González-Rubio et al., 2010]. However, it must be studied how to obtain the most profit of them.

A.5 Evaluating the IMT prototype

The MIPRCV project web based demo featured a big amount of these interesting features, namely confidence measures in the translated words, mouse click operations, and electronic pen interaction. We will refer to this system as the advanced demonstrator (IMT-AD, Figure A.4a). The addition of such advanced features conditioned the design of the interface; e.g., the use of a text field for each word eased dramatically the e-pen interaction, at the expenses of an unusual text flow, and thus, a keyboard interaction a bit different from typical text areas.

The goal of the first evaluation was aimed to assess both qualitatively and quantitatively IMT-AD, and compare it to a state-of-the-art post-editing (PE) MT

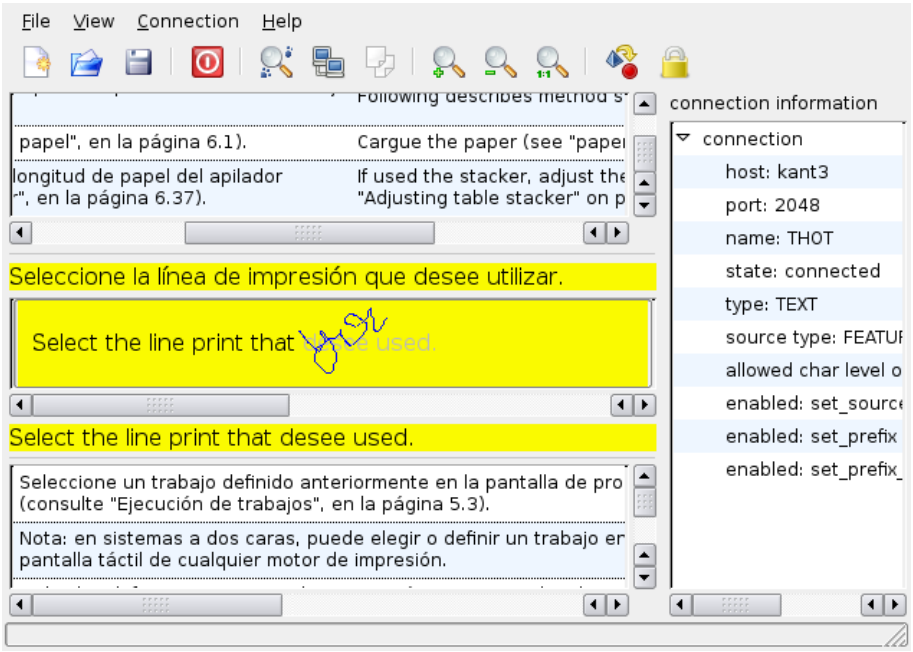


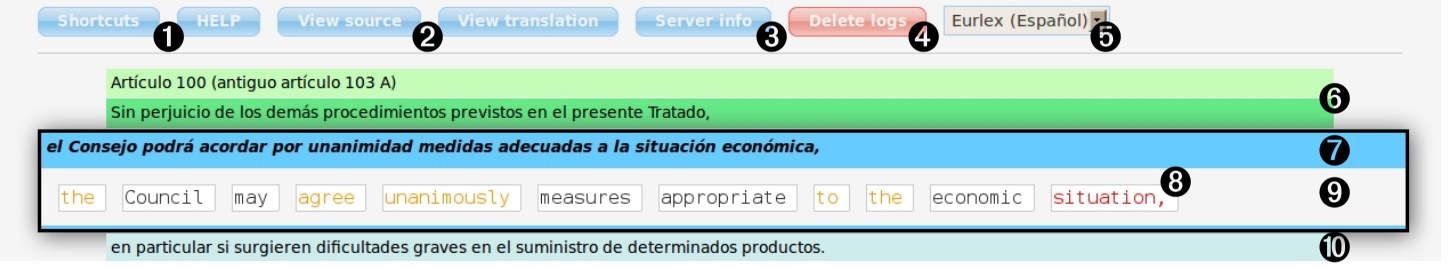
Figure A.3: Screen shot of the installable application when using pen-stroke corrections.

output. Translating from scratch was not considered since it is becoming obsolescent in the translation industry except for the case that computer assisted tools cannot be used. However, PE of MT systems is becoming more prevalent and it can be found quite frequently in a professional translation workflow [Carl, 2012; SchlumbergerSema S.A. et al., 2001]. Thus, in addition to IMT-AD, a post-editing version of the demonstrator (PE-AD) was developed to make a fair comparison with state-of-the-art PE systems. PE-AD used the same interface as IMT-AD, but the IMT engine was replaced by autocompletion-only capabilities as popular word processors have.

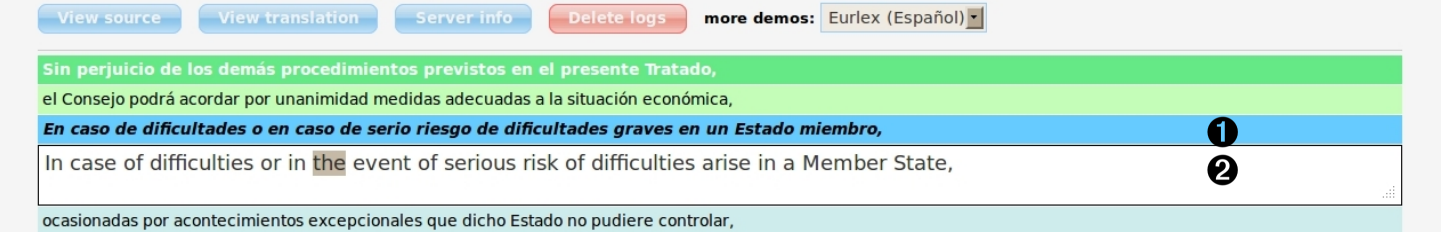
Design

Both systems were evaluated on the basis of the ISO 9241-11 standard (ergonomics of human-computer interaction). The following three aspects were considered: efficiency, effectiveness, and user satisfaction.

Firstly, efficiency should be measured in a way that the translation industry understands well, e.g., the number of words per hour. However, it would also be interesting to know the amount of time the user was not interacting with the system, which can be an indicator of the user's cognitive effort. Secondly, the effectiveness measures the quality of the resulting translation. As human



(a) Detail of the advanced web-based interface. At the top, the menu includes buttons to show the help and shortcuts ❶, buttons to show the original document and a composed version of the output document ❷, a button to show the status of the server ❸, a button to clear all the translation in the page ❹ and a drop-down list to change the corpus ❺. Below, in green the already translated sentences ❻, in blue the source sentence being translated ❼, and at the bottom, in light blue the source sentences to be translated ❿. The working are is shown in the middle of the page in a gray background ❾. Inside, the word-boxes ❽ are highlighted with orange if the confidence of the system that the word is correct is middle, and in red if the system has low confidence.



(b) Detail of the simplified web-based interface. This interface basically mimics the advanced interface except for ❶ and ❷ which are now implemented in HTML+javascript instead of Adobe Flash. As a result, ❷ has been simplified and the boxes in the advanced interface have been replaced by an HTML text area. Thus, confidence measures and e-pen interaction are not allowed in this prototype.

Figure A.4: Details of the advanced and simplified web-based interfaces.

evaluation of the final translations can be time and budget costly, the automatic evaluation of the final translations can be measured with the residual error. Ideally, both human and automatic evaluation should give the same results. Although it is known that they do not are equivalent, they are correlated to some extent [Papineni et al., 2002]. However, this is a pessimistic approach since we have just one reference, but there may be many good translations. Finally, we need to measure user satisfaction by assessing human judgment of the proposed systems and measuring system usability, which, in turn, can be leveraged as a means of feedback to improve the system in future developments.

Hence, we decided to approach these concerns with the following solutions. For the former, we computed the average time in seconds that took to complete each translation. For the second, we evaluated the BLEU against the reference and a crossed multi-BLEU among users' translations. Finally, we adapted the system usability scale (SUS) questionnaire to score the user satisfaction, by asking 10 questions that users would assess in a 1–5 Likert scale (1:strongly disagree, 5:strongly agree), plus a text area to submit free-form comments.

A form was designed to asses user satisfaction. It consisted of 10 questions in a likert scale[2] plus a text area to fill with personal comments (Figure A.5). The 10 questions were the following:

Q1 I think I would like to use the IMT application frequently
Q2 I found the IMT system unnecessarily complex
Q3 I think that the IMT application is easy to use
Q4 I think I would need technical assistance to use this system
Q5 I find the different features of the IMT application well embedded
Q6 I think that the IMT system presents some inconsistency
Q7 I imagine most people would learn to use the IMT application quickly
Q8 I find the IMT system cumbersome to use
Q9 The use of the IMT system is reliable
Q10 I would need to learn much before using the IMT system

Participants

A group of 10 users (3 females) aged 26–43 from our research group volunteered to perform the evaluation as non-professional translators. All of them were proficient in Spanish and had an advanced knowledge of English. Although none had worked with IMT systems, all knew the basis of the IMT paradigm.

Apparatus

Since participants were Spanish natives, we decided to perform translations from English to Spanish. We chose a medium-sized corpus, the EU corpus,

[2] http://core.ecu.edu/psyc/wuenschk/StatHelp/Likert.htm

Figure A.5: Picture of questionnaire for the advanced prototype with the 10 likert questions ❶ and the text area for free comments ❷.

typically used in IMT experiments [Barrachina et al., 2009]. It consists of documents from the European Union. We built a glossary for each source word by using the 5-best target words from a word-based translation model (Figure A.6). We expected this would cover the gap of knowledge for this particular task of our non-expert translators. In addition, a set of 9 keyboard shortcuts was designed, aiming to simulate a real translation scenario, where

the mouse is typically used sparingly. The list of shortcuts is the following:

- insert after current word (`CTRL+SPACE`)
- insert before current word (`CTRL+SHIFT+SPACE`)
- delete following word (`CTRL+SUPR`)
- reject current hypothesis (`click` or `CTRL+UP`)
- obtain previous hypothesis (`CTRL+DOWN`)
- validate current word (`ENTER`)
- validate the sentence (`CTRL+ENTER`)
- validate the sentence up to the current word (`CTRL+SHIFT+ENTER`)

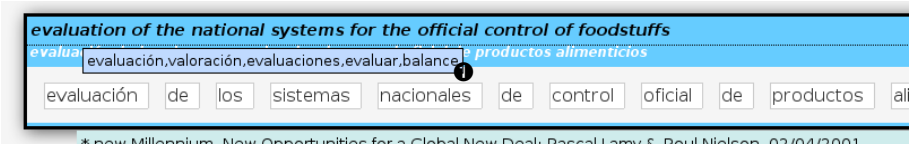


Figure A.6: Details of the glossary. A glossary is shown as a tooltip box with the 5 best translations from the statistical dictionary ❶.

Furthermore, autocompletion was added to PE-AD, i.e., words with more than 3 characters were autocompleted using a task-dependent word list. In addition, IMT-AD was set up to predict at character level interactions. We decided to disable the e-pen interaction and confidence measures, so that way we could focus on keyboard-only IMT evaluation.

Procedure

Three disjoint sentence sets (C1, C2, C3) were randomly selected from the test dataset. Each set consisted of 20 sentence pairs and kept the sequentiality of the original text. Sentences longer that 40 words were discarded. C3 was used in a warm up session, where users gained experience with the IMT system (5–10 min per user on average) before carrying out the actual evaluation. Then, C1 and C2 were evaluated by two user groups (G1, G2) in a counterbalanced fashion: G1 evaluated C1 on PE-AD and C2 on IMT-AD, while G2 did C1 on IMT-AD and C2 in PE-AD.

Results

Although the results were not conclusive (there were no statistical differences), the results showed some trends for this particular group of users. First, the time spent (efficiency) in the IMT system (67s per sentence average) was higher than in PE (62s per sentence average). However, the effectiveness was slightly higher for IMT in BLEU with respect to the reference (41.5 vs 40.7) and with respect to a cross-validation with other user translations (78.9 vs 77.4). This

	PE-AD	**IMT-AD**
Avg. time (s)	62 ($SD = 51$)	67 ($SD = 65$)
BLEU	40.7 (13.4)	41.5 (13.5)
Crossed BLEU	77.4 (4.5)	78.9 (4.8)
Satisfaction		
Q1	2.5(1.1)	2.4(1.0)
Q2	2.2(1.2)	2.5(1.0)
Q3	3.5(1.1)	3.1(1.2)
Q4	1.9(1.0)	1.9(0.9)
Q5	3.2(1.0)	3.1(1.2)
Q6	2.8(1.0)	3.6(0.9)
Q7	3.3(1.1)	3.4(1.0)
Q8	2.8(0.7)	3.5(0.8)
Q9	3.6(0.8)	2.8(0.6)
Q10	1.7(1.0)	1.9(0.9)
Global satisfaction	2.5(1.2)	2.1(1.2)

Table A.1: Summary of the results for the first test.

suggested that the IMT system helped to achieve more consistent and standardized translations. Finally, the PE system was perceived more satisfactory than the IMT system. In fact, the global satisfaction score was around 2.5 for PE and 2.1 for IMT on a scale of 5, which suggested that users were not comfortable with none of the systems, especially with the IMT system. This result was discouraging. However, it points out the directions to take in order to improve the usability of the IMT system. In particular, IMT obtained 3.6 in **Q6** while PE obtained 2.8, while in **Q8** they obtained 3.5 and 2.8 respectively, and in **Q9** 2.8 and 3.6 respectively. This means that the IMT system was considered more inconsistent, more cumbersome and less reliable than the PE system.

This was corroborated by the user comments who complained about too many shortcuts and edit operations available, some operations not working as expected, the word-box based interface, and some annoying common mistakes in the predictions of the IMT engine (e.g., inserting a whitespace instead of completing a word, which would be interpreted as two different words). One user stated that the PE system "was much better than the [IMT] predictive tool". With respect to the PE system, the users basically complained about the autocompletion not being useful. A summary of the results can be found in Table A.1.

A.5.1 Simplified Web Based Prototype

The results from the first evaluation were quite disappointing. Not only participants took more time to complete the evaluation with the IMT system, but also they perceived that IMT-AD was more cumbersome and unreliable than PE-AD. However, we still observed that IMT-AD was being beneficial in some circumstances, and probably the bloated UI was the cause for IMT to fail. Thus, we decided to develop a simplified version of the original prototype (Figure A.4b).

Design

In the simplified prototype, the word-box based interface was changed to a simple text area. Consequently, confidence measures and multimodality are no longer available. However, this was not a problem since they had been already disabled in the first evaluation. In addition, the edit operations were simplified to allow only word substitutions and single-click rejections, which match better the theoretical IMT protocol established in the automatic evaluation. Besides, we expected that the simplification of the interface logic would reduce some of the programming bugs that bothered users in the first evaluation. The PE interface was simplified in the same way. Furthermore, the autocompletion feature was improved to support n-grams of arbitrary length.

Participants

Fifteen participants aged 23–34 from university English courses (levels B2 and C1 from the Common European Framework of Reference for Languages) were paid to perform the evaluation (5 € each). A special price of 20 € was given to the participant who would contribute with the most useful comments about both prototypes. It was found that, following this method, participants were more verbose in their comments and suggestions.

Apparatus

In this case, a different set of sentences (C1′, C2′, C3′) was randomly extracted from the EU corpus. The sentences were filtered so that the average complexity (measured in WSR) was similar to the average complexity of the whole corpus. Moreover, the sentences with complex formating, like bullets or unusual amount of non-character symbols, were also filtered out.

Procedure

To avoid the bias regarding which system was being used, sentences were presented in random order, and the type of system was hidden to the participants. As a consequence, users could not evaluate each system independently. Therefore, a reduced questionnaire with just two questions was shown on a

	PE-BD	IMT-BD
Avg. time (s)	69 ($SD = 42$)	55 ($SD = 37$)
No. interactions	94 (60)	79 (55)
Avg. backward cursor moves	19 (15)	15 (15)
Q1 (Likert scale)	3.1 (1.2)	3.5 (1.1)
Q2 (Likert scale)	2.9 (1.2)	3.1 (1.3)

Table A.2: Summary of results for the second test.

per-sentence basis. **Q1** asked if the system suggestions were useful. **Q2** asked if the system was cumbersome to use. A text area for free-form comments was also included.

Results

Still with no statistical significance, we found that the IMT prototype was perceived now better than PE for this particular group of users (Table A.2). First, interacting with IMT was more efficient than with PE on average (55 s vs. 69 s). The number of interactions was also lower (79 vs. 94). We also observed that users went back to rectify parts of the prefix that had been already corrected (15 backward cursor moves). Concerning user satisfaction, the IMT system was perceived as more helpful (3.5 vs. 3.1) but also more cumbersome (3.1 vs. 2.9). However, in this case the differences were narrower. On the other hand, we performed a manual sentiment analysis, where IMT received 16 positive comments whereas PE received only 5. Regarding negative comments, the counts were 35 (IMT) and 31 (PE). While the number of negative comments is similar, there was an important difference regarding the positive ones.

Finally, the users complaints of the IMT system can be summarized in the following items:

1. system suggestions changed too often, offering very different solutions;
2. while correcting one mistake, subsequent words that were correct were changed by a worse suggestion;
3. system suggestions did not keep gender, number, and time concordance;
4. if the user goes back in the sentence and performs a correction, parts of the sentence already corrected were not preserved on subsequent system suggestions.

A.6 Summary of contributions

Our initial UI performed poorly when tested with real users. However, when the UI design was adapted to the users' expectations, the results were encouraging.

Note that in both cases the same IMT engine was evaluated under the hood. This fact remarks the importance of the UI design when evaluating a highly interactive system as is IMT.

The literature had reported good experimental results in simulated-user scenarios, where IMT is focused on optimizing some automatic metric. However, user productivity is strongly related to how the user interacts with the system and other UI concerns. For instance, a suggestion that changes on every key stroke might obtain better automatic results, whereas the user productivity decreases because of the cognitive effort needed to process those changes. Therefore, a new methodology is required for optimizing interactive systems (like IMT) towards the user.

As a summary, the following issues need to be addressed in an IMT system:

1. user corrections should not be modified, since that causes frustration;
2. system suggestions should not change dramatically between interactions, in order to avoid confusing the user;
3. the system should only propose a new suggestion when it is sure that it improves the previous one.

We hope these considerations will reduce the gap between human translators and technology, so that future developments can have an impact on the translation industry.

The results of this evaluation lead to the following publication:

- **V. Alabau**, L. A. Leiva, D. Ortiz-Martínez, and F. Casacuberta. User Evaluation of Interactive Machine Translation Systems. In *Proc. of the 16th Annual Conference of the European Association for Machine Translation (EAMT'12)*, p. 20–23, 2012.

Bibliography

V. Alabau, D. Ortiz, V. Romero, and J. Ocampo. A Multimodal Predictive-Interactive Application for Computer Assisted Transcription and Translation. In *Proceedings of the ICMI-MLMI 2009*, pp. 289–292, 2009.

V. Alabau, J. Benedi, F. Casacuberta, L. Leiva, D. Ortiz-Martinez, V. Romero, J. Sanchez, R. Sanchez-Saez, A. Toselli, and E. Vidal. CAT-API Framework Prototypes. In *Proceedings of Database and Expert Systems Applications (DEXA), 2010 Workshop on Interactive Multimodal Pattern Recognition in Embedded Systems (IMPRESS 2010)*, pp. 264–265, 2010a.

V. Alabau, F. Casacuberta, L. Leiva, D. Ortiz-Martínez, and G. Sanchis-Trilles. Sistema web para la traducción automática interactiva. In *Actas del XI Congreso Internacional de Interacción Persona Ordenador. INTERACCIÓN2010*, pp. 47 – 56, 2010b.

V. Alabau, D. Ortiz-Martínez, A. Sanchis, and F. Casacuberta. Multimodal Interactive Machine Translation. In *ICMI-MLMI '10: Proceedings of the 2010 International Conference on Multimodal Interfaces*, 2010c.

V. Alabau, A. Sanchis, and F. Casacuberta. Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 389–394, 2011.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. L. Lagarda, H. Ney, J. Tomás, E. Vidal, and J. M. Vilar. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.

M. Carl, editor. *International Workshop on Expertise in Translation and Post-editing Research and Application (ETP 2012)*, 2012.

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.

J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta. On the Use of Confidence Measures within an Interactive-predictive Machine Translation System. In *Proceedings of 14th Annual Conference of the European Association for Machine Translation*, 2010.

P. Koehn. Enabling Monolingual Translators: Post-Editing vs. Options. In *Proc. ACL-HLT*, 2010.

A. Lagarda, L. Rodríguez, E. Cubel, E. Vidal, and F. Casacuberta. Transtype 2. Un sistema de ayuda a la traducción. In *Proceeding of the SEPLN: XIX Congreso de la Sociedad Española para el Procesamiento del Lenguaje*, pp. 345–346, 2003.

P. Langlais, G. Lapalme, and M. Loranger. TRANSTYPE: Development-Evaluation Cycles to Boost Translator's Productivity. *Machine Translation*, 15(4):77–98, 2002.

M. Luján-Mares, V. Tamarit, V. Alabau, C.-D. Martínez-Hinarejos, M. Pastor, A. Sanchis, and A. Toselli. iATROS: A speech and handwritting recognition system. In *V Jornadas en Tecnologías del Habla (VJTH'2008)*, pp. 75–78, 2008.

D. Ortiz, L. A. Leiva, V. Alabau, and F. Casacuberta. Interactive Machine Translation using a Web-based Architecture. In *Procedings of the International Conference on Intelligent User Interfaces*, pp. 423–425. 2010.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

V. Romero, L. A. Leiva, A. H. Toselli, and E. Vidal. Interactive Multimodal Transcription of Text Images Using a Web-based Demo System. In *Procedings of the International Conference on Intelligent User Interfaces*, pp. 477–478, 2009.

V. Romero, A. H. Toselli, and E. Vidal. *Multimodal Interactive Handwritten Text Transcription*. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing, 2011.

R. Sánchez-Sáez, L. A. Leiva, J.-A. Sánchez, and J.-M. Benedí. Interactive Predictive Parsing using a Web-based Architecture. In *Proceedings of the Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'10)*, pp. 37–40, 2010.

G. Sanchis-Trilles, M. González, F. Casacuberta, E. Vidal, and J. Civera. Introducing Additional Input Information into IMT Systems. In *Proceedings of the 5th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms. Lecture Notes in Computer Sciences*, volume 5237, pp. 284–295. 2008a.

G. Sanchis-Trilles, D. Ortiz-Martínez, J. Civera, F. Casacuberta, E. Vidal, and H. Hoang. Improving Interactive Machine Translation via Mouse Actions. In *EMNLP 2008: conference on Empirical Methods in Natural Language Processing*, 2008b.

SchlumbergerSema S.A., I. T. de Informática, R. W. T. H. A. L. f. I. VI, R. A. e. L. I. L. U. of Montreal, C. Soluciones, S. Gamma, and X. R. C. Europe. TT2. TransType2 - Computer Assisted Translation. Project Technical Annex., 2001.

List of Figures

List of Tables


structures
sentence
position
needed
proposed
good
test
goal
scenario
MT
cases
previous
on-line
element
list
function
elements
sum
addition
find
labels
fact
respect
perform
space
source
text
algorithm
rule
strategies
result
context
assume
automatic
decoding
prediction
posterior
output
model
statistical
strategy
computed
contextual
training
recognition
results
system
errors
systems
optimum
active
approach
words
probabilities
search
techniques
hypothesis
error
word
data
corpus
models
algorithms
probability
confidence
AISP
WER
ISP
based
time
suffix
prefix
defined
input
found
improve
similar
WG
level
MAP
loss
number
target
problems
SISP
account
process
structured
PR
set
obtained
ASR
samples
sequence
HTR
hand
case
label
real
Note
state
IMT
high
reduce
final
PE
order
task
feedback
multimodal
Correction
interaction
speech
e-pen
keyboard
supervision
translation
cost
effort
mouse
decision
problem
quality
transcription
corrections
solution
translations
user
reference
post-editing
handwriting
correct
correcting
users
language
handwritten
amend
supervise
work
human
SP
shown
obtain
structure
experiments
tasks
provide