UNIVERSITAT POLITÈCNICA DE VALÈNCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Aplicaciones de los autómatas transductores finitos con pesos (WFST) en la corrección simbólica en interfaces persona-máquina

Tesis presentada por: José Ramón Navarro Cerdán
Dirigida por: Rafael Llobet Azpitarte
Joaquim Arlandis Navarro
Juan Carlos Pérez Cortés

Para la obtención del título de Doctor en Informática

Diciembre 2015

ÍNDICE GENERAL

LISTA DE ACRÓNIMOS

FSA autómata de estados finitos 31

FST autómata transductor de estados finitos 31

WFST autómata transductor de estados finitos con pesos 31

OCR reconocimiento óptico de caracteres 36

SR reconocimiento automático del habla 36

HM modelo de hipótesis 36

EM modelo de error 36

CM modelo de restricciones 36

IM modelo de interacción 37

EM$_{IM}$ modelo de error asociado a la interacción con el usuario 45

EM$_{HM}$ modelo de error asociado al modelo de hipótesis 46

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Las interfaces persona-máquina de entrada de datos, independientemente de cuál sea su naturaleza: texto manuscrito, voz, teclado, etc., están sujetas a una gran variedad de errores. La combinación de determinadas fuentes de información, producida durante el propio proceso de introducción de datos, puede contribuir de manera significativa en la reducción de dichos errores. En la tesis que aquí se presenta, dicha fusión de la información se va a abordar mediante el uso de autómatas transductores de estados finitos (WFST) y la operación de composición. Esta metodología permite el modelado de cada una de las fuentes de información de manera independiente. El resultado final de la composición de éstos autómatas genera un único autómata que integra todas las fuentes de información previamente modeladas. Como ejemplo de uso de la metodología propuesta, se presentan diferentes aplicaciones relativas a la corrección simbólica en interfaces persona-máquina, tanto desde un enfoque automático como desde el enfoque de la ayuda a la interacción. Los resultados se presentan bajo la perspectiva de un postproceso de cadenas provenientes de un proceso de digitalización de formularios.

Por otra parte, disponer de la posibilidad de discernir las cadenas correctamente postprocesadas de las incorrectas es un tema de enorme importancia. Es por ello que se plantea un método general de estimación del error frente a un coste de transformación de las cadenas de entrada que permite establecer un umbral dinámico en función de dicho coste y un parámetro propuesto por el usuario final: el error final asumible.

En esta tesis se presenta inicialmente una aplicación real de corrección de cadenas procedentes de un clasificador OCR en una tarea de digitalización de formularios. Estas cadenas, proceden de un clasificador con cierta probabilidad de error, lo que implica la posibilidad de que alguno de los caracteres pertenecientes a una palabra sea erróneo, produciendo finalmente palabras incorrectas. Esto plantea la necesidad de introducir algún tipo de postproceso que mejore dichas cadenas. Para implementar dicho postproceso, se tienen en cuenta todas las evidencias disponibles en un momento dado. En el caso propuesto aquí serán los caracteres reconocidos por el propio clasificador con su probabilidad a posteriori, la matriz de confusión entre símbolos y el modelo de lenguaje finalmente aceptado. Cada una de estas evidencias

se modela de manera independiente en forma de un WFST. Una vez modeladas se fusionan mediante la operación de composición de autómatas en un único autómata integrado. A partir de este autómata, se selecciona el camino que maximiza la probabilidad y que corresponde con la cadena perteneciente al lenguaje más cercana a la hipótesis OCR según la matriz de confusión entre símbolos. El sistema final ofrecerá dos resultados diferentes: por una parte la cadena corregida y por otra el coste de transformación de dicha corrección.

Dado que el postproceso descrito está sujeto a la generación de posibles transcripciones erróneas, en muchos casos puede ser conveniente acotar el error final producido. Ello conduce a la aplicación de políticas de rechazo de hipótesis en base a un umbral de confianza (o, equivalentemente, un umbral sobre el coste de transformación). En esta tesis se presenta un método adaptativo de estimación del umbral de rechazo que permite estimarlo para obtener un determinado porcentaje de error en un lote de cadenas de un lenguaje (muestra) que presenta diversas ventajas. Por un lado, es independiente de la distribución de los costes de transformación de dichas muestras. Por otro lado, permite al usuario establecer el umbral de una manera familiar y ventajosa, como es fijando la tasa de error deseada de la muestra. Para todo ello, en primer lugar, y para un lenguaje dado, se define un modelo que estima la probabilidad de error asociada a aceptar cadenas con un coste de transformación determinado. A continuación, se expone el procedimiento que lleva a cabo la estimación del umbral de rechazo de manera adaptativa con el objetivo de alcanzar la tasa de error predefinida para un lote de test. Además, se propone una aproximación para la obtención del modelo anterior cuando no se dispone de hipótesis OCR reales y supervisadas en la etapa de aprendizaje. El capítulo se acompaña de experimentos cuyos resultados demuestran la utilidad del método propuesto.

Seguidamente y enlazando en cierta forma con la búsqueda de un incremento de productividad en una posible validación de las cadenas, previamente rechazadas por el sistema a través del método de estimación del error anteriormente expuesto, se presenta un método de interacción persona-máquina multimodal e interactivo que fusiona la información anterior junto al prefijo introducido, por el propio usuario, durante dicho proceso de validación, haciendo uso para ello de los WFST y la operación de composición de autómatas. La búsqueda de la cadena más probable, para cada nueva interacción ofrecida por el usuario, en el autómata compuesto que aquí se expone, muestra un claro incremento de la productividad, al requerirle un número menor de pulsaciones de teclado en la obtención de la cadena correcta. Para finalizar, se muestra otra interfaz multimodal e interactiva tolerante a fallos, mediante la fusión de diferentes fuentes de información junto a un modelo de error relacionado con las posibles confusiones producidas debido a la disposición de las teclas de un teclado. Para ello, se hace uso también de WFST para su modelado. La aplicación mostrada en este caso está relacionada con la introducción de un destino en un dispositivo GPS y en ella se considera, tanto la información de los destinos próximos a un lugar concreto, como la información relativa al prefijo introducido y los errores que pueden aparecer debido a la propia disposición de las teclas en el dispositivo de entrada.

RESUM

Les inferfícies persona-màquina d'entrada de dades, independentment de quina siga la seua naturalesa: text manuscrit, veu, teclat, etc., estan subjectes a una gran varietat d'errors. La combinació de determinades fonts d'informació, produïda durant el procés d'introducció de dades, pot contribuir de manera significativa a la reducció d'aquests errors. A la tesi que ací es presenta, la fusió d'informació s'ha dut a terme mitjançant l'ús d'autòmats transductors d'estats finits (WFST) i l'operació de composició. Aquesta metodologia permet el modelat de cadascuna de les fonts d'informació de manera independent. El resultat final de la composició d'aquests autòmats genera un únic autòmat que integra totes les fonts d'informació prèviament modelades. Com a exemple d'ús de la metodologia proposada, es presenten diferents aplicacions relatives a la correcció simbòlica en interfícies persona-màquina, tant des d'un punt de vista automàtic com des del punt de vista de l'ajuda a la interacció. Els resultats es presenten sota la perspectiva d'un postprocés de cadenes provinents d'un procés de digitalització de formularis.

D'una altra banda, disposar de la possibilitat de distingir les cadenes correctament postprocesades de les incorrectes és un tema important en un procés de postprocés automàtic i és per això que es planteja un mètode general d'estimació de l'error front al cost de transformació de les cadenes d'entrada que permet establir un llindar dinàmic en funció d'aquest cost i un paràmetre proposat per l'usuari final: l'error final assumible.

En aquesta tesi es presenta inicialment una aplicació real de correcció de cadenes procedents d'un classificador OCR en una tasca de digitalització de formularis. Aquestes cadenes, procedeixen d'un classificador amb una determinada probabilitat d'error, la qual cosa implica la possibilitat de que algun dels caràcters que pertanyen a una paraula siga erroni, produint finalment paraules incorrectes. Això planteja la necessitat d'introduir algun tipus de postprocés que millore aquestes cadenes. Per implementar aquest postprocés, es tenen en compte totes les evidències disponibles en un moment donat. En el cas proposat ací, seran els caràcters reconeguts pel propi classificador amb la seua probabilitat a posteriori, la matriu de confusió entre símbols i el model de llenguatge finalment acceptat. Cadascuna d'aquestes evidències es modela de manera independent en forma d'un WFST. Una vegada modelades

es fusionen mitjançant l'operació de composició d'autòmats en un únic autòmat integrat. A partir d'aquest autòmat, es selecciona el camí que fa màxima la probabilitat i que es correspon amb la cadena més propera a la hipòtesi OCR que pertany al llenguatge segons la matriu de confusió entre símbols. El sistema final oferirà dos resultats diferents: d'una banda la cadena corregida, i d'una altra, el cost de transformació d'aquesta correcció.

Atés que el postprocés que es descriu està subjecte a la generació de possibles transcripcions errònies, en molts casos pot ser convenient delimitar l'error final produït. Tot això condueix a la aplicació de polítiques de rebuig d'hipòtesis prenent com a base un llindar de confiança (o de manera equivalent, un llindar sobre el cost de transformació). En aquesta tesi es presenta un mètode adaptatiu d'estimació de rebuig, amb la finalitat d'obtindre un determinat percentatge d'error en un lot de cadenes d'un llenguatge (mostra) que presenta diversos avantatges. D'una banda és independent de la distribució dels costos de transformació de les mostres esmentades. D'altra banda, permet l'usuari establir el llindar d'una manera familiar i avantatjosa, com és fixant la tasa d'error desitjada per la mostra. Per tot això, en primer lloc, i donat un llenguatge, es defineix un model que estima la probabilitat d'error associada a acceptar cadenes amb un cost de transformació determinat. A continuació, s'exposa el procediment que du a terme l'estimació del llindar de rebuig de manera adaptativa amb l'objectiu de arribar a la tasa d'error predefinida per a un lot de test. A més a més, es proposa una aproximació per a obtindre el model anterior quant no es disposa d'hipòtesi OCR reals i supervisades a l'etapa d'aprenentatge. El capítol s'acompanya d'experiments, els resultats dels quals demostren la utilitat del mètode proposat.

Seguidament, i enllaçant amb la recerca d'un increment en la productivitat en una possible validació de cadenes prèviament rebutjades pel sistema a través del mètode d'estimació de l'error anteriorment exposat, es presenta un mètode d'interacció persona-màquina multimodal i interactiu que fusiona la informació anterior, juntament amb el prefix introduït pel propi usuari durant l'esmentat procés de validació, fent ús dels WFST i l'operació de composició d'autòmats. La recerca de la cadena més probable, en cada nova interacció oferida per l'usuari ens mostra un clar increment de la productivitat, al requerir un nombre menor de pulsacions de teclat per obtindre la cadena correcta. Per finalitzar, es mostra una altra interfície multimodal i interactiva tolerant a errades, mitjançant la fusió de diferents fonts d'informació juntament a un model d'error relacionat amb les possibles confusions produïdes a causa de la disposició de les lletres d'un teclat. En aquest cas es fa ús també dels WFST en el seu modelat. L'aplicació mostrada en aquest cas està relacionada amb la introducció d'una destinació en un dispositiu GPS i en aquesta es considera tant la informació pròxima a un lloc concret, com la informació relativa al prefix introduït, junt als errors que poden aparèixer a causa de la pròpia disposició de les tecles en el dispositiu d'entrada.

SUMMARY

Human-machine or data entry interfaces, irrespective of their nature: handwritten, voice, keyboard, etc., are subject to a variety of errors. The combination of several information sources, some with origin in the the data entry process itself, can significantly contribute to reduce those errors. This thesis shows the fusion of that information by using Weighted Finite State Transducers (WFST) and the composition operation. This methodology allows us to model each information source independently. The final composition of these automata gives rise to a global transducer that integrates all information sources that were previously independently modelled. As an example of the proposed methodology, different Human-machine Interface applications related to symbolic correction and focused on automatic correction and interaction support are shown. Results are presented under the perspective of digitizing forms process.

Additionally, the capability to discern the correctly postprocessed strings from those that are still incorrect is very important. That is why it is proposed a general method of error estimation using the input string transformation cost that establishes a threshold in terms of the cost and the proposed end-user parameter: the acceptable final error.

In this thesis a real application related to the string correction process from an OCR classifier in a form digitizing task is presented. These strings come from a classifier with a given error ratio that implies that some characters in the string have been potentially misclassified, producing erroneous words. This raises the need to introduce some kind of postprocess to improve the strings. The implementation of such postprocess takes into account all the available evidence in a given moment. In the case proposed here, these evidences are the characters recognized by the classifier with their posterior probabilities, the confusion matrix between symbols and the language model finally accepted. Each evidence is modelled independently by means of a WFST and then combined by means of the composition operation into a single integrated automata. From this automata, the path that maximizes the probability is selected. This path is the string, that belongs to the language model, that is the nearest string to the

OCR hypothesis according to the confusion matrix. The final system offers two different results: on the one hand the corrected string, on the other hand the transformation cost produced during the string correction.

Since the postprocessing described is subject to possible erroneous generation transcripts, in many cases it may be desirable to limit the final error occurred. This leads to the implementation of policies of rejection of hypotheses based on a confidence threshold (or, equivalently, a threshold on the cost of processing). This thesis presents a method for estimating adaptive rejection threshold estimation that allows for a certain percentage of error in a lot of strings from one language (sample) that presents several advantages. On the one hand, it is independent from transformation cost postprocessing distribution of such samples. On the other hand, it allows the user to set the threshold for a familiar and advantageous manner, as is setting the desired rate of sampling error. For this, first, and for a given language, a model that estimates the probability of error associated with the acceptation of postprocessed strings with a given transformation cost is defined. Then, the procedure that performs the rejection threshold estimation adaptively in order to achieve predefined rate error for a test batch is presented. In addition, an approach to obtain the above model is proposed when there are no real and supervised OCR hypothesis in the learning stage. The chapter is accompanied by experiments whose results demonstrate the utility of the proposed method. Next, linking in somehow with the search for an increased productivity in a possible string validation task, of previously strings rejected by the system through the foregoing error estimation method, a method of multimodal and interactive human-computer interaction that composes the above information with the prefix introduced by the user, while the validation process occurs, making use, for this, of WFST and the automata composition operation. The search for the most likely string for each new interaction offered by the user, in the composed automata, presented here, shows a clear increase in productivity by requiring fewer keystrokes in obtaining the correct string. Finally, a tolerant fault multimodal and interactive interface, using also WFST, is shown by making the composition of different information sources together with an error model related with the possible confusion caused due to the arrangement of keys on a keyboard. The application shown in this case is related to the introduction of a destination into a GPS device where is considered both the information related to the next destinations to a specific place, such as the information related to the entered prefix and errors that may occur due to the arrangement of keys on the input device considered.

RECONOCIMIENTOS

Son muchas las ocasiones en que vienen a mi memoria momentos felices de otra época, la de mi infancia. Es en esa época donde se van dibujando poco a poco las metas que uno desea alcanzar a lo largo de su vida, aunque quizás algo desdibujadas y sin una hoja de ruta clara para poder llevarlas a cabo. El esfuerzo y el paso del tiempo hace que se vayan consiguiendo pequeños hitos que van conformando las posibles opciones sobre las que posteriormente se toman determinadas decisiones que van creando los surcos de los caminos siguientes a tomar. Atrás he dejado mis tiempos de la escuela elemental, en la que crecí, mi feliz paso por el bachillerato, y como no, mi paso por la facultad, donde me formé personal y profesionalmente. Tras este periodo de formación vino mi experiencia durante unos años en el mundo de la empresa y mi vuelta, quizás por casualidad, al mundo académico, donde me especialicé en el Reconocimiento de Formas y el Análisis Estadístico. Es aquí donde comencé mi carrera docente e investigadora y también es en estos últimos años en los que, como consecuencia de esta inquietud investigadora, he dedicado una gran parte de mi tiempo libre en esa carrera de fondo que es el desarrollo de la Tesis. Esto ha supuesto un gran esfuerzo tanto personal como familiar que no puedo dejar de agradecer a todos los familiares, compañeros y amigos a los que de alguna forma les pudiera también haber afectado.

La elaboración de esta tesis ha pasado por varias fases, desde las iniciales en las que lo que se hace parece no llevarte al fin buscado, hasta las últimas donde todo finalmente toma forma y sorprendentemente cuadra con la mayor parte de las expectativas iniciales e incluso en algún caso las mejora. El paso por las distintas fases no es un camino fácil y requiere de mucha perseverancia y apoyo. Es por ello que quiero agradecer el apoyo prestado por parte de mis directores: Rafael Llobet Azpitarte, Joaquim Arlandis Navarro y Juan Carlos Pérez Cortés, que durante el paso por éstas han sabido guiar y formar parte en todo momento del desarrollo del trabajo que ahora se presenta. También tengo que agradecer la contribución de Vicent Castelló Fos en el desarrollo de la interfaz gráfica web que muestra el funcionamiento de una de las partes de la tecnología que aquí se muestra y a Alejandro Héctor Toselli por su ayuda en los aspectos relativos al formato de la tesis.

CAPÍTULO 1

PROBLEMÁTICA, OBJETIVOS Y ESTADO DEL ARTE

Índice del capítulo

1.1. Problemática

Los excelentes resultados que los humanos obtienen a la hora de leer un texto manuscrito se debe mayoritariamente a su extraordinaria habilidad de recuperación de errores, gracias, sobre todo, a la aplicación constante de restricciones léxicas, sintácticas, semánticas, pragmáticas y discursivas del lenguaje. Cualquier método de procesamiento automático de la entrada de datos va a estar sujeta a una enorme variedad de errores e incertidumbre por lo que se hace necesaria la aplicación de algoritmos de corrección que permitan postprocesar dicha entrada.

Formalmente, el papel de un sistema de interacción en la entrada, es el de maximizar la probabilidad de que una cadena, recibida como hipótesis procedente de diferentes subsistemas de entrada multimodales sea correcta, en el sentido de que sea compatible con las restricciones de entrada impuestas por la tarea (lenguaje). Estas restricciones conforman el Modelo de Restricciones o Modelo de Lenguaje y pueden ser tan simples como un pequeño conjunto de palabras válidas, (por ejemplo posibles valores de una provincia en un campo de un formulario, o una ciudad en un sistema de navegación), o tan complejo como una frase cualquiera en lenguaje natural.

Todas estas aproximaciones toman como entrada una cadena propuesta por el subsistema de símbolos de entrada, aplican un Modelo de Lenguaje, y con frecuencia optimizan una transformación de costes utilizando un Modelo de Error, pero, por lo general, no toman en consideración otras fuentes de entrada o información probabilística relativa a los diferentes símbolos que conforman dicha cadena que correspondería al *Modelo de Hipótesis*. Dependiendo del subsistema de símbolos de entrada, este *Modelo de Hipótesis* puede incluir *probabilidades a posteriori* u otros índices de fiabilidad de los símbolos más probables. Otro elemento a tomar en consideración es la información referente a la confusión entre símbolos, modelada a través de la Matriz de Confusión, cuya información puede ser incluida en lo que se conoce como *Modelo de Error*.

Por otra parte, y como tema transversal al de corrección de cadenas, en algunos sistemas, tras esta corrección aparece la necesidad de estimación del error cometido. En dichos sistemas, con objeto de incrementar la productividad durante el proceso de validación, en el caso de requerirse ésta, o simplemente con objeto de ofrecer un determinado nivel de calidad en las cadenas finalmente ofrecidas, se establece un umbral de rechazo sobre las cadenas corregidas (o transcripciones) a través de alguna de las medidas que estos sistemas de corrección ofrecen, de forma que todas las cadenas que no superen cierto valor umbral de coste son aceptadas como correctas, mientras que aquéllas que lo superan son clasificadas como incorrectas. Las cadenas rechazadas pueden pasar a un proceso de evaluación humana que suele hacer uso de algún tipo de ayuda interactiva durante el proceso de validación. Es evidente que el establecimiento de dicho umbral va a tener implícito un error debido a que no todas las cadenas aceptadas como buenas van a ser correctas (falsos positivos), ni tampoco todas las cadenas rechazadas van a estar mal corregidas (falsos negativos).

Si se requiere de un proceso de validación de las cadenas rechazadas, entonces, frecuentemente, se requerirá de algún tipo de interfaz persona-máquina que permita dicha validación. Las interfaces persona-máquina suelen estar sujetas a gran cantidad de errores e incertidumbre. Esto hace necesaria la aplicación de modelos de interacción con objeto de mejorar la

velocidad y calidad de la misma, minimizando para ello los errores de la entrada. En cualquier sistema de introducción de datos, desde un teclado a un reconocedor OCR, pasando por interfaces táctiles, navegadores, etc., los campos de entrada o elementos de diálogo suelen tener una sintaxis del lenguaje y una semántica conocidas previamente, además de otras posibles fuentes de información como la disposición de las teclas en el teclado, o un sistema de sonido o nuevos sensores como acelerómetros o sistema de geolocalización. La combinación de tales fuentes de información puede ayudar, tanto a la reducción del esfuerzo requerido por el humano durante la validación, como a la reducción de los errores finales producidos por éste.

1.2. Introducción

Una interfaz persona-máquina está frecuentemente sujeta a gran cantidad de errores e incertidumbre, por ello es de vital importancia la aplicación de un modelo de postproceso que permita la corrección de dichos errores. El objetivo de tal modelo consiste en la mejora de la velocidad y la calidad de la interacción minimizando los errores de entrada. En muchos sistemas de entrada de datos, ya sea un teclado software convencional o un reconocedor OCR, incluyendo interfaces táctiles, dispositivos industriales, navegadores GPS, sistemas de control, etc., se conoce previamente la sintaxis y la semántica del lenguaje de los diferentes campos o elementos de diálogo que conforman dicha entrada de datos. Cuando los elementos de entrada tienen una complejidad y variabilidad inherente, como así ocurre, en el reconocimiento de caracteres y gestos, o cuando los subsistemas de entrada tienen limitaciones ergonómicas debidas al tamaño, al peso, a la forma o a limitaciones en el manejo, como en un teléfono móvil, teclados táctiles, etc., el uso de un método sofisticado de corrección de símbolos de entrada o postproceso, probablemente mejorará la eficiencia para el usuario. Otro elemento que aparece frecuentemente en los sistemas actuales es la combinación de varias fuentes de entrada como pantallas táctiles, un teclado, un sistema de entrada de sonido, nuevos sensores como acelerómetros, GPS, etc. El tener en cuenta toda esta variabilidad multimodal de información, conlleva la necesidad de introducción de metodologías que permitan fusionar las diferentes fuentes de información.

En esta tesis se propone un sistema automático de corrección, y dos sistemas interactivos de ayuda a la interacción humana. Ambos sistemas combinan varias fuentes de información, incluyéndose también como fuente la evidencia probabilística, con objeto de tratar dos problemas diferentes: el primero sería la obtención de la cadena más probable a partir de un modelo de hipótesis, un modelo de error y un modelo de restricciones con el objetivo de corregir las cadenas de entrada incorrectas y transformarlas en cadenas de salida válidas desde el punto de vista de la tarea; el segundo problema a tratar consistiría en implementar la corrección interactiva y multimodal en el caso de que la cadena requiera una validación humana, ejemplo de éstas son tareas de digitalización de formularios en el que el resultado final no debe superar un error determinado y por lo tanto se requiere de un proceso de validación de aquellas cadenas de menor calidad que superaban el umbral de rechazo. Sin embargo, existen tareas en las que la validación humana no es necesaria, bien porque el propio usuario sabe si la cadena requiere o no de corrección, un ejemplo de estas tareas es la entrada de un teclado

de un dispositivo móvil o dispositivo de navegación GPS, o bien porque tan solo se requiere trabajar con las cadenas cuya calidad es aceptable pudiéndose rechazar el resto. Ejemplo de este tipo de tareas puede ser la clasificación de documentos.

Las fuentes de información de las que habitualmente se dispone son: a) la hipótesis de entrada, que puede ser tan simple como una secuencia de símbolos o tan compleja como un grafo representando un conjunto de frases probabilísticas que pertenecen a una gramática. Esto es lo que nosotros llamamos *Modelo de Hipótesis (HM)*. b) Un modelo de errores esperados en la hipótesis de entrada junto a sus probabilidades, a esto le llamamos el *Modelo de Error (EM)* y habitualmente dicha información viene a través de matrices de confusión entre símbolos, y c) el lenguaje al cual pertenecen las cadenas de entrada, a este modelo le vamos a llamar *Modelo de Restricciones (CM)*. En muchos trabajos este modelo coincide con el *Modelo de Lenguaje (LM)*. Si todos estos modelos se combinan de manera apropiada, el sistema propondrá la cadena más probable compatible con el CM.

Para aquellas tareas que requieran de un proceso de supervisión humana previo a la fase de corrección interactiva y multimodal, se ha de detectar el conjunto de cadenas sobre las que dicha fase debe actuar. Para ello, el primer paso consiste en calcular el coste de corrección de una cadena, dicho coste se ofrece directamente tras el postproceso. Se supone que a mayor coste de corrección mayor esfuerzo realizado y por lo tanto mayor probabilidad de que dicha cadena sea incorrecta. Bajo ese supuesto se estima un modelo que permite relacionar los costes de corrección con la probabilidad de error y a través de éste y el error asumible por el usuario se es capaz de establecer un umbral que permite la aceptación/rechazo de las cadenas pertenecientes a un lote. Para la realización de este modelo se aplican técnicas de histograma y ventanas móviles que permiten buscar la relación entre los costes de corrección de las cadenas y la probabilidad de error para cada coste en cada uno de los modelos de restricciones. Esto nos permitirá establecer de manera automática el umbral a partir del cual se garantiza el error asumible por el usuario. Así, si la cadena propuesta no es la esperada, se debe permitir la interacción con el usuario para la corrección del error. Esta interacción nos lleva a una nueva fuente de información que vamos a denominar *Modelo de Interacción con el Usuario (IM)*. Este nuevo modelo puede combinarse dinámicamente (según el usuario vaya interactuando con el sistema) junto con los otros modelos, para obtener la salida deseada con el mínimo esfuerzo por parte de éste.

Es posible incluso, modelar el hecho de que el usuario pueda cometer errores mientras interactúa con el sistema (por ejemplo tecleando una tecla adyacente a la tecla verdadera). En este caso el Modelo de Interacción con el Usuario puede tener asociado su propio Modelo de Error que permita recuperarse ante la introducción de errores en la propia entrada de éste. De hecho el modelo IM puede ser visto como un segundo Modelo de Hipótesis, el cual lleva a un sistema de corrección de errores interactivo e incluso multimodal (si las fuentes de interacción proceden de varios frentes como por ejemplo, teclado, voz, etc), donde varios símbolos de entrada (hipótesis) se combinan para proponer la salida. En la aproximación propuesta los distintos modelos de información se representan mediante transductores (WFST), en adelante entenderemos la acción de transducir como la operación de transformar unos símbolos de entrada en otros de salida por medio de un transductor. En la figura 1.1 se propone un sistema de corrección de errores interactivo y multimodal, donde el operador $\circ$ representa la operación

de composición de autómatas. En un primer paso, se componen los modelos HM, EM y CM, lo que permite la transducción de una cadena $x \in L_{\text{HM}}$, donde L_{HM} sería el lenguaje generado por HM, de acuerdo con las operaciones de error definidas en EM (asumiendo que EM transduce cualquier cadena en cualquier otra). En la fase de decodificación se encuentra la más probable de estas transducciones. Si el usuario no está satisfecho con la salida propuesta, entonces se permite la interacción para mejorar el resultado de la cadena. En este caso, se crea un transductor de estados finitos con pesos (WFST), llamado IM, componiéndose con HM∘EM lo que finalmente impone nuevas restricciones al sistema.

En la aproximación que se presenta no se han hecho asunciones sobre la procedencia de las hipótesis iniciales ni sobre cómo el usuario interactúa con el sistema. La hipótesis inicial puede ser la salida de un sistema OCR, una entrada por teclado, por voz, o cualquier otra interfaz persona-máquina o cualquier combinación de varias de ellas. La interacción del usuario puede proceder de cualquier medio físico, ya sea software (basado en pantallas táctiles), reducido (teclado de teléfono móvil) o incluso por medio de reconocedores de gestos, de voz, OCR, etc.

La idea es representar la información presente en cada modelo, (HM, EM, CM, IM) por medio de transductores de estados finitos (WFST), y componerlos todos juntos con objeto de encontrar la cadena más probable de entre todas las presentes en el modelo CM, de acuerdo a la hipótesis actual presente en HM y teniendo en cuenta, además, los posibles errores modelizados en EM, junto a la entrada producida por el usuario que está modelizada en IM. Este problema puede ser resuelto si se componen en el orden adecuado todos estos modelos y se busca el camino más probable en el autómata resultante final tras la aplicación de la operación de composición de todos ellos.

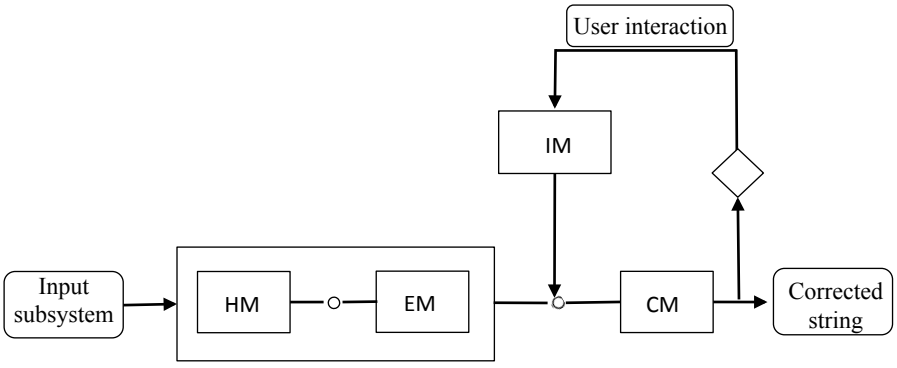

Figura 1.1: Esquema de un postproceso interactivo multimodal.

1.3. Objetivos

Esta tesis pretende mostrar la enorme eficiencia y flexibilidad que ofrecen los autómatas transductores finitos con pesos (WFST) en la modelización de interfaces interactivas. Para

ello se va a mostrar su potencial en una tarea de corrección automática de cadenas tras un reconocimiento OCR. Tras este reconocimiento las cadenas serán aceptadas o rechazadas en función del error final que el usuario es capaz de asumir para una tarea dada. Para ello, se plantea además un modelo general de estimación del error, que relacione dicho error con un umbral establecido sobre un coste de corrección. Para acabar, se muestra la flexibilidad de los WFST a la hora de modelizar la información referente al proceso de interacción del usuario durante la validación humana, además de su eficiencia a través del incremento de productividad que la inclusión de información interactiva y multimodal ofrece.

1.3.1. Objetivos generales

En esta tesis se plantean dos objetivos principales bien diferenciados:

1. *Corrección automática de cadenas mediante el uso de WFST.*

Las aportaciones que se realizarán en este campo son: por una parte el uso de la composición, junto a otro tipo de operaciones con autómatas, que permitan la integración de diferentes fuentes de información relacionadas con la corrección automática de cadenas. Para ello se combinarán modelos de restricciones o de lenguaje, probabilidades a posteriori del subsistema de entrada y un modelo de error entre símbolos. También se aborda, la problemática sobre la estimación del error en función de los costes de corrección como medida de estimación de la calidad. Los resultados obtenidos en este estudio son de importancia pues permiten ofrecer automáticamente un umbral de dicho coste a partir del error asumido por el usuario. Mediante el método de estimación expuesto en este capítulo, el sistema será capaz, de manera general, de estimar el umbral del coste que garantice un error dado para cada lenguaje y de manera independiente.

2. *Mejoras en interfaces multimodales e interactivas persona-máquina haciendo uso de WFST.*

En la segunda parte, se presenta una metodología basada en WFST que permite fusionar la información disponible. Como ejemplos de fuentes de información cabe resaltar la información procedente de la interacción con el usuario, las características de los dispositivos de entrada, la ubicación del usuario o el punto de destino, en el caso por ejemplo de interacción con un dispositivo GPS, etc. Las aportaciones en este campo estarán relacionadas con el incremento de productividad que se obtiene al realizar la combinación de dichas fuentes de información en un sistema interactivo. Para ello se plantean dos aplicaciones prácticas que permiten ilustrar la bondad de la metodología propuesta: primero un sistema de postproceso de cadenas interactivo que nos va a ser útil a la hora de estudiar el incremento de la productividad que se produce durante la validación humana de aquellas cadenas que no han sido aceptadas como válidas de manera automática por el sistema planteado en el primer punto, y una segunda aplicación, donde se estudia el efecto de la fusión de información de naturaleza distinta, (hipótesis de entrada + localización GPS), a partir de la información procedente de

un navegador GPS, junto con el modelo interactivo del usuario, con el objetivo final de mostrar la capacidad de integración de información de naturaleza diversa y multimodal mediante este tipo de autómatas y mostrar además el incremento de productividad que este tipo de información adicional puede aportar en el ejemplo de búsqueda de un destino en un navegador GPS.

1.3.2. Objetivos específicos

Objetivo 1: Corrección automática de cadenas mediante el uso de WFST

En los procesos de digitalización de formularios suelen buscarse mejoras en la productividad a través de una reducción del esfuerzo necesario para realizar dicha digitalización. Para conseguir esto, se utilizan técnicas como la que se propone en este punto, que permiten corregir las cadenas reconocidas de manera automática en cadenas válidas de un lenguaje dado. Existe un doble objetivo al aplicar estas técnicas, por una parte se pretende minimizar el error final con la mejora de la cadena que este proceso de corrección produce, y por otra parte se pretende incrementar la productividad al aceptar de manera automática las cadenas correctas y que por tanto no deberán de pasar por un proceso de validación humana posterior, con el incremento de productividad que ello conlleva. Para ello es necesario el uso de restricciones y conocimiento a priori sobre el contenido de los campos. En el caso de campos de un formulario, las restricciones más frecuentes son léxicas (sólo se acepta un número limitado de posibles cadenas) o sintácticas (se incluye redundancia o códigos de control en la identificación o en los códigos de número de pasaporte o cuentas bancarias, etc.). En este punto se va a hacer uso de la operación de composición sobre autómatas transductores finitos estocásticos, abordando dos tipos de mejoras, inclusión de las probabilidades a posteriori de un clasificador e interdependencia de campos, que producen una reducción considerable del error en el reconocimiento de cadenas. Además, como un tema transversal pero que nos sirve como vínculo de unión entre el objetivo 1 y el objetivo 2 se va a presentar un método general que permite establecer de modo automático un umbral de aceptación de cadenas dado un error final asumible por el usuario. Los subobjetivos que se plantean referentes a este primer objetivo son:

Postproceso del reconocimiento OCR de campos aislados usando WFST y probabilidades a posteriori del clasificador Aquí se hace uso de la flexibilidad de los WFST para incluir el conjunto de todas las clasificaciones ofrecidas por el clasificador OCR dando el peso correspondiente a cada una de ellas mediante las probabilidades a posteriori.

Postproceso del reconocimiento OCR haciendo uso de la interdependencia de campos mediante WFST En muchas ocasiones existe un grado de interdependencia entre campos de un formulario. Para aprovechar esta interdependencia se definen los modelos de lenguaje combinados. En este apartado se han creado y probado los modelos WFST y se reportan las mejoras observadas.

Estimación del umbral de rechazo para una muestra de cadenas asumiendo un error predefinido A partir de una cadena de un campo de formulario reconocido por el clasificador OCR, el sistema propone la cadena más probable perteneciente al lenguaje asociado, junto con el coste de la corrección. Se propone, a partir de una muestra de entrenamiento supervisada: 1) establecer los modelos apropiados que relacionen dicho coste con el error cometido, y 2) formular un método para la estimación automática de un umbral de rechazo que permita al usuario definir el error esperado para el conjunto de cadenas aceptadas (no rechazadas) de una muestra completa, independientemente de su distribución de costes (es decir, muestras de entrada de diferente dificultad de reconocimiento).

Metodología para la estimación de la curva de error frente al coste de corrección en modelos sin muestras reales supervisadas En muchas ocasiones aparece la necesidad de crear un modelo de lenguaje de corrección completamente nuevo. Ante un modelo así nos encontramos con el problema de no poseer muestras reales que nos permitan estimar directamente la curva de error frente al coste de corrección. Se pretende realizar dicha estimación a partir de las propias cadenas que forman el lenguaje, y la información de confusión entre caracteres, mediante la generación de muestras sintéticas.

Objetivo 2: Mejoras en las interfaces multimodales e interactivas persona-máquina mediante el uso de WFST

Los procesos de interacción persona-máquina suelen ser procesos costosos cuando la única fuente de información que se considera es la propia interacción. Ejemplos de procesos de interacción humana pueden ser: corregir una cadena, escribir un mensaje en un móvil, buscar una ubicación en un GPS, etc. Si en un proceso se dispone de otro tipo de fuentes de información como pueden ser: el modelo de restricciones o de lenguaje al que pertenecen las cadenas de un campo en un formulario, la lengua a la que pertenece el mensaje que se está escribiendo en un mensaje de móvil, la disposición y tamaño de las teclas en un teléfono móvil, una idea aproximada de un lugar de destino en un mapa (basándonos por ejemplo en los destinos más recientes, en el número de habitantes de un municipio o en la proximidad del destino a un área previamente seleccionada), etc. En estos casos el coste del proceso de interacción puede verse muy reducido si somos capaces de integrar la información disponible en dicho proceso.

La idea que se persigue consiste en incrementar la productividad en la corrección humana de cadenas reduciendo el número de teclas que se haya de pulsar hasta obtener la cadena correcta, y esto se acometerá mediante la fusión de toda la información presente en el proceso de validación. Para ello se hará uso de los WFST a la hora de combinar las diferentes fuentes de información. En este caso se va a tomar como fuentes de información de entrada: las probabilidades a posteriori de un clasificador de caracteres, el modelo de restricciones (modelo de lenguaje) de las cadenas que son admisibles en un campo, el modelo de error entre caracteres y el prefijo de la cadena que el validador humano va introduciendo y que corresponde con la parte interactiva del modelo planteado. Para cada letra pulsada el sistema irá produciendo una corrección diferente compatible con la parte de la cadena introducida por el usuario y

con el resto de información de la que ya se disponía. En este segundo objetivo se presentan los siguientes subobjetivos:

Postproceso OCR interactivo En los primeros dos subobjetivos anteriores se planteaba el postproceso automático de cadenas OCR, sin intervención humana, mediante el uso de autómatas WFST, tras este postproceso y a partir de la información procedente del método de estimación del error en función del coste de corrección planteado en los últimos dos subobjetivos, se enviarán a validación una serie de cadenas para cumplir con el error final asumido por el usuario. Lo que se pretende aquí es incrementar la productividad durante dicho proceso de validación de la cadena sin renunciar a la fiabilidad final del sistema, para ello se hará uso de los WFST y a la información anterior se le adicionará la información procedente del prefijo *online* que se va introduciendo durante dicho proceso de corrección manual. Se evalúa el incremento de productividad y fiabilidad producido al adicionar diferentes informaciones durante dicho proceso de validación.

Uso de los WFST para la combinación de información interactiva multimodal En este punto se estudiará la flexibilidad de los WFST a la hora de añadir información interactiva multimodal y se evaluará cómo dicha información incrementa el rendimiento del sistema con costes temporales de respuesta muy bajos, evaluándose además el efecto de la introducción de la posibilidad de tolerancia a fallos.

1.4. Aportaciones

En la presente tesis se presenta una metodología que permite la fusión de diferentes fuentes de información y que está basada en autómatas WFST y la operación de composición de autómatas. Esta metodología se muestra de manera incremental tomando como ejemplo una tarea de digitalización de formularios donde se distinguen tres fases bien diferenciadas. Primero se fusiona la información con el objetivo de obtener una corrección automática de las cadenas procedentes de un clasificador OCR. Tras esto se establece un modelo capaz de relacionar el coste de corrección de la cadena con el error asociado a ésta lo que permitirá establecer un umbral de rechazo de cadenas en función del error asumido por el usuario. Para finalizar se vuelve a hacer uso de los WFST y la operación de composición de autómatas para facilitar la interacción humana durante el proceso de corrección de las cadenas rechazadas. Así pues las aportaciones que se presentan en esta tesis son:

- Un método flexible de fusión de la información para la corrección automática de cadenas basado en la composición de autómatas WFST. En él se muestra la metodología seguida y se comparan los resultados obtenidos con la aplicación de otro tipo de postproceso basado también en autómatas pero que no hace uso de los autómatas transductores y la operación de composición. También se muestran las mejoras obtenidas al añadir la información de las probabilidades a posteriori. Por otra parte se expone la manera de modelar la información combinada procedente de varios campos, cuando estos estén relacionados, y los resultados obtenidos al hacer uso de tal información

combinada respecto a cuando la corrección se realiza de manera aislada sin aprovecha dicha información de contexto. Relacionado con este tema se han presentado las publicaciones:

1. *OCR Post-processing Using Weighted Finite-State Transducers*
2. *Efficient OCR Post-Processing Combining Language, Hypothesis and Error Models.*
3. *Using Field Interdependence to Improve Correction Performance in a Transducer-based OCR Postprocessing System*

junto al registro de la propiedad intelectual: *Biblioteca de funciones itilmc2 para la corrección contextual de cadenas*.

- Metodología para la estimación de un umbral de rechazo de cadenas adaptativo a partir del error asumido por el usuario. En esta parte se propone y evalúa una nueva metodología para la estimación adaptativa del umbral de rechazo a partir del error asumido por el usuario, comparándose esta nueva metodología con la umbralización fija clásica. Relacionado con este tema se han presentado las publicaciones:

  1. *Batch-adaptive rejection threshold estimation with application to OCR post- processing*
  2. *Rejection Threshold Estimation for an Unknown Language Model in an OCR Task*
  3. *User-defined expected error rate in OCR postprocessing by means of automatic threshold estimation*

- Método de interacción multimodal a partir de la fusión de información de la que se disponga sobre las diferentes partes que conforman nuestro entorno de interacción además de la propia interacción ofrecida por el usuario. Se muestra y evalúa el uso de de los WFST y la operación de composición en dos ejemplos distintos. En un primer ejemplo se trata de mejorar la interacción del usuario, reduciendo el esfuerzo requerido por éste a la hora de corregir las cadenas rechazadas por el método de umbralización anteriormente expuesto. En un segundo ejemplo, no relacionado ya con la corrección de cadenas OCR, se muestra una nueva aplicación relativa a la búsqueda de un destino en un dispositivo GPS en el que adicionalmente a la propia interacción del usuario durante su proceso de elección del destino deseado se tienen en cuenta la información multimodal sobre la ubicación actual además de evaluar el comportamiento de dichos modelos con y sin tolerancia a fallos durante el propio proceso de entrada de datos. Relacionado con este tema se han presentado las publicaciones:

  1. *Improvement of Embedded Human-Machine Interfaces Combining Language, Hypothesis and Error Models*
  2. *Composition of Constraint, Hypothesis and Error Models to improve interaction in Human-Machine Interfaces*

  y la patente: *Method for Symbolic correction in human-machine interfaces*.

1.5. Plan de la obra

En esta tesis sobre corrección simbólica en interfaces persona-máquina mediante el uso de WFST se han buscado una serie de aplicaciones sobre las que ejemplificar diferentes usos y rendimientos de este tipo de autómatas en lo referente a la mejora de cadenas. Para ello se plantea una tarea inicial, con bastante interés en la industria de la digitalización de formularios, y que está enfocada en la mejora de cadenas procedentes de un clasificador OCR. A partir de este planteamiento inicial surgen una serie de necesidades posteriores que se van resolviendo en los diferentes capítulos, bien mediante la aplicación de otro tipo de técnicas o bien mediante la inclusión de nuevas evidencias modeladas también por medio de WFST. Así pues, en el capítulo 1 que ahora nos ocupa se introduce la problemática y se ofrece una visión sobre el estado del arte de las diferentes partes que en esta tesis se muestran. Seguidamente en el capítulo 2 se presenta el marco matemático en el que se fundamentan los WFST además de la operación de composición de autómatas que es la operación principal utilizada en el uso de los autómatas planteado en ésta tesis y algunas de sus propiedades estructurales a nivel probabilístico. Tras esto, en el capítulo 3 se muestra la metodología a seguir mediante una descripción de los distintos tipos de autómatas WFST que se van a ir introduciendo en las diferentes partes del trabajo presentado conforme estos vayan siendo requeridos, además de mostrar el modelo probabilístico que estos modelan.

En el capítulo 4 se parte de la información obtenida de un clasificador OCR que nos ofrece una clasificación multiclase y se plantea la aplicación de una nueva metodología, capaz de combinar fácilmente diferentes fuentes de información que permitan la corrección automática de cadenas. En este tipo de clasificadores cada clase tiene un peso determinado en función de las características presentes en la imagen. Normalmente este peso es directamente, o puede transformarse fácilmente, en una probabilidad a posteriori de las clases. En el apartado 4.3 se muestra la flexibilidad de los WFST a la hora de combinar e introducir diferentes fuentes de información. Concretamente en este apartado, se muestra una aplicación que introduce la información relativa a las probabilidades a posteriori que el clasificador OCR ofrece. En los WFST que aquí presentamos, dichas probabilidades a posteriori se encuentran en las aristas de transición entre estados dentro de un autómata. Las aristas además contienen como información los símbolos de entrada y salida a las que ellas mismas representan. Así pues, la información que encontramos en cada una de las aristas nos dice que transducir el símbolo de entrada en el símbolo de salida tiene el peso que viene indicado en ella. En este apartado, además de mostrar la facilidad a la hora de introducir dicha información, se hace una comparativa que muestra las mejoras ofrecidas por la información que aportan dichas probabilidades a posteriori del clasificador. Por otra parte, se compara también con un método de postproceso de cadenas anterior que no hacía uso de las WFST y todo ello se compara finalmente con el resultado final que se obtendría en las salidas de un OCR si estas no fuesen postprocesadas. En ocasiones, existen campos dentro de un formulario que tienen cierto grado de interdependencia, entendiendo como tal interdependencia el hecho de que el conocimiento del valor de uno de los campos reduce el número de posibilidades que pueden tomar los restantes. Esto redunda en una menor entropía a la hora de realizar el postproceso. Ejemplos típicos de este tipo de campos son: regiones, estados, provincias, ciudades, calles, códigos postales, etc. En

el apartado 4.4 se van a aprovechar los WFST para introducir dicho grado de interdependencia entre campos de un formulario. Para sacar partido de esta interdependencia se definen los modelos de lenguaje combinados, que hacen uso de un carácter comodín que será un carácter fijo, que no aparezca en ninguno de los campos con interdependencia y que servirá de unión entre todos ellos. A este carácter le exigiremos que no se le pueda someter a ningún tipo de transformación, (sustitución, inserción, borrado), y nos será de gran utilidad en la delimitación de cada uno de los campos que forman parte de esta combinación. Esto nos permitirá, entre otras cosas, poder asignar el resultado final de la etapa de postproceso, a los respectivos campos dentro de un formulario. Operando de esta manera es como si el conjunto de campos con interdependencia fuese uno único. Los resultados finales que se muestran aquí consisten en una comparativa entre la corrección de dichos campos de forma aislada y la corrección de forma combinada.

En las empresas de digitalización masiva de formularios, existen muchas ocasiones en las que se admite la posibilidad de cometer un pequeño porcentaje de error final. Este error es el error máximo que la empresa cliente es capaz de asumir a la hora de aceptar un lote como bien digitalizado. El sistema, que se muestra en el capítulo 4, nos ofrece la mejor cadena perteneciente a un lenguaje, a partir del reconocimiento producido junto con su coste de corrección, el cual está relacionado con el esfuerzo que se ha tenido que hacer durante dicho proceso de corrección automática para convertir la salida del OCR en la cadena más próxima que pertenezca al modelo de lenguaje o de restricciones (CM) establecido. Dicho coste se usa finalmente para establecer el umbral a partir del cual una cadena es aceptada automáticamente como buena, o por el contrario, pasa a un subproceso de validación humana. Cabe destacar aquí que entre las cadenas aceptadas habrá cadenas erróneas que serán las que conformen el error final asumido, entendiéndose pues, que las cadenas rechazadas que pasan a validación acaban corrigiéndose correctamente. El problema aquí reside en que mientras que el error asumido si que es un concepto natural directamente conocido por el usuario, el coste sin embargo, es un concepto que depende mucho del modelo de lenguaje que se aplique en un campo lo que dificulta enormemente su establecimiento. En el capítulo 5, se propone y se evalúa una metodología general que permite relacionar directamente el error esperado a partir del coste de corrección para cada modelo de lenguaje. La finalidad de esta estimación no es otra que la de ser capaces de establecer un umbral automático, a partir del error que el usuario es capaz de asumir. Todo esto se lleva a cabo gracias al conocimiento de esta relación en cada uno de los lenguajes. Así pues, dicho umbral, calculado indirectamente a través del error asumido por el usuario, será la frontera entre las cadenas reconocidas automáticamente que se dan como buenas, y que por lo tanto no han de pasar por un proceso de validación humana, con el incremento de productividad que ello conlleva, de aquellas cadenas que necesariamente han de ser validadas por un humano en la tarea de digitalización de documentos.

En el caso de tener la necesidad de corregir las cadenas rechazadas anteriormente, se requiere de un proceso de validación humana que se realiza habitualmente mediante la presentación al validador humano de la imagen de la cadena dudosa junto con un campo donde aparece el texto reconocido. Tras esto, el humano decide si lo reconocido automáticamente por el sistema es correcto, mediante la única pulsación de una tecla en el caso de que la cadena postprocesada se corresponda con la real, o bien transcribe completamente la cadena que aparece en la imagen cuando ésta no ha sido correctamente postprocesada. En el capítulo

6, y concretamente en el subapartado 6.5.1, se va a mostrar la capacidad de los WFST a la hora de combinar gran variedad de fuentes de información de índole diversa, con el fin de reducir el esfuerzo necesario en el proceso de corrección humana de cadenas al disminuir el número de teclas necesarias que éste ha de pulsar hasta obtener la cadena correcta. Esto se acometerá mediante la fusión de la información disponible y relevante para dicho proceso de validación. En este caso se va a tomar como fuentes de información de entrada: las probabilidades a posteriori de un clasificador de caracteres, el modelo de restricciones (modelo de lenguaje) de las cadenas que son admisibles en un campo, el modelo de error entre caracteres y el prefijo de la cadena que el validador humano va introduciendo y que corresponde con la parte interactiva del modelo planteado. Para cada letra pulsada el sistema irá produciendo una corrección diferente compatible con la parte de la cadena introducida por el usuario y con el resto de información de la que ya se disponía.

Por otra parte, también en este capítulo, y con objeto de mostrar el potencial de los WFST a la hora de introducir información multimodal, se presenta un problema en el que se dispone de diversas fuentes de este tipo de información, ejemplo de la cual podemos encontrar en los dispositivos móviles actuales que permiten tanto una entrada tecleada como hablada o los actuales GPS que permiten conocer el destino a través de sus teclas, a través de la voz o incluso podrían hacerlo pulsando simplemente sobre la ubicación aproximada de destino dentro de un mapa mostrado en pantalla. Debido a las reducidas dimensiones que estos dispositivos suelen poseer o las situaciones de entorno en las que se requiere de dicha introducción de datos, cualquier combinación de información que ayude a incrementar la productividad es útil. En el subapartado 6.5.2 se muestra un ejemplo que consiste en mostrar el potencial de los WFST a la hora de incrementar la productividad a través de la reducción de errores y del número de pulsaciones de teclas necesarias en la introducción de datos en dichos dispositivos. Para ello se estudiará el efecto de la introducción de la información disponible de localización de un navegador GPS, junto con el modelo interactivo del usuario sobre dispositivos móviles actuales, además de evaluar también el efecto de introducción de la tolerancia a fallos durante el propio proceso de entrada.

Para finalizar, en el capítulo 7 se presentan las conclusiones de cada una de las diferentes partes del trabajo, además del trabajo futuro y las publicaciones generadas a partir del desarrollo de esta tesis.

La última parte de la tesis es el anexo A donde se han ubicado los detalles más técnicos que no resultaban relevantes para la comprensión de los objetivos perseguidos pero que parecía importante destacar de cara a una comprensión más detallada y profunda de estas partes.

1.6. Estado del arte

Cualquier interfaz para la interacción, conocida también como *interfaz de lenguaje* puede ser modelada a diferentes niveles Berghel [1987]. El más bajo de todos los niveles es el nivel de palabra. Éste envuelve restricciones léxicas sobre la secuencia de símbolos que hay en el interior de la unidad más pequeña de interfaz semántica, y que habitualmente es lo que

denotamos como palabra. El siguiente, es el nivel de frase; un campo en un formulario o una línea de entrada en una caja de diálogo, son ejemplos de este nivel. Los modelos a nivel de palabra y de frase suelen aplicar métodos de búsqueda en diccionarios, n-gramas, técnicas basadas en distancias de edición, modelos ocultos de Markov y otros modelos de transición entre categorías de palabras o caracteres. Los niveles más altos consideran contextos más amplios que requieren de conocimiento a priori del dominio de la aplicación.

El problema de corrección de palabras en el texto se focaliza habitualmente en la detección de palabras fuera del vocabulario, la corrección de palabras aisladas y la corrección de palabras a nivel de frase. Para la resolución de cada uno de estos problemas se aplican diferentes técnicas relacionadas con el reconocimiento de formas y el modelado del lenguaje natural. En Kukich [1992] se presenta un estudio sobre patrones de errores ortográficos y técnicas utilizadas en la detección de palabras fuera del vocabulario y corrección de errores en palabras aisladas, ofreciendo además una revisión del estado del arte de diferentes técnicas utilizadas en la corrección de palabras dependientes del contexto.

Los métodos más simples a la hora de realizar el postproceso de una cadena, utilizan un léxico de palabras conocidas y preguntan a un operador que verifique o introduzca manualmente los símbolos de entrada de las palabras desconocidas. Se pueden utilizar técnicas específicas a la hora de realizar una búsqueda aproximada en un léxico. En Hall and Dowling [1980] se presenta un excelente artículo sobre métodos de búsqueda de cadenas, donde entre otras aparecen técnicas que se basan en búsquedas en vecinos más cercanos sobre determinados espacios métricos, considerando las cadenas como puntos en un espacio con disimilitudes y realizando técnicas de reducción de la dimensionalidad, búsqueda en árboles, o búsquedas rápidas basadas en las propiedades de las métricas. Existen otros métodos que se basan en n-gramas o en máquinas de estados finitos Berghel [1987]; Farooq et al. [2009]; Perez-Cortes et al. [2000]; T Breuel [1994], en los que una cadena candidata es analizada y el conjunto de transiciones con costes más bajos (mayor probabilidad) define la cadena de salida. El algoritmo de Viterbi Amengual and Vidal [1998]; Neuhoff [1975], es el algoritmo clásico más ampliamente utilizado a la hora de buscar el camino de máxima probabilidad sobre una máquina de estados finitos con objeto de realizar el análisis y corrección de errores sobre una gramática regular.

La mayoría de los trabajos en el modelado de lenguaje han sido llevados a cabo en el campo del reconocimiento automático del habla Jelinek [1991]. Aunque los requerimientos son distintos, muchas de las técnicas básicas usadas en esta disciplina son exportables a otro tipo de tareas con pequeñas modificaciones. El modelado que se aplica tanto a nivel de palabra como de frase se basa en métodos de búsqueda en diccionarios, *n-gramas*, Modelos Ocultos de Markov y técnicas basadas en distancias de edición. Existen varios trabajos que utilizan técnicas de modelado de lenguaje, en entornos con y sin restricciones, con el objeto de aplicar la corrección del error procedente de tareas dedicadas a OCR o reconocimiento de texto. Algunos de estos ejemplos los podemos encontrar en Hull and Srihari [1982]; Kolak and Resnik [2005]; Perez-Cortes et al. [2000]; Tong and Evans [1996].

En Taghva and Stofsky [2001] se presenta un sistema para la corrección de errores ortográficos generados a partir de la digitalización de documentos. En este trabajo se seleccionan

las palabras candidatas a través del uso de información procedente de varias fuentes de conocimiento y sus correcciones se basan en técnicas estadísticas, correspondencias aproximadas de cadenas y *n-gramas*. El sistema es capaz de aprender la información de confusión a partir de muestras obtenidas de una colección de documentos. Hassan Awadallah et al. [2008] hace uso de autómatas de estados finitos (FSA) a la hora de proponer correcciones candidatas con una distancia de edición específica con las correcciones con errores ortográficos. Al Azawi and Breuel [2014] hace uso de autómatas transductores de estados finitos (WFST) para modelizar la información de confusiones contextuales de símbolos obtenidas a partir del algoritmo Levenshtein [1966] y fusionar dicha información con la información procedente de un OCR y el modelo de lenguaje de salida ambos modelizados también mediante autómatas WFST.

La fusión de las diferentes evidencias que se disponga en un momento dado, puede contribuir significativamente en la mejora de las tasas de reconocimiento final. Así, en Meyer et al. [2004] describe un sistema en el que se fusiona información audiovisual en un sistema automático de reconocimiento de voz. En este sistema toma en consideración el sonido junto con las características visuales de la boca al pronunciar las palabras con el objetivo de reducir el error en el reconocimiento de palabras. En Khaleghi et al. [2013] se presenta un excelente revisión del estado del arte sobre la fusión de datos, revisándose desde las diferentes problemáticas encontradas hasta metodologías aplicadas ofreciéndose además una visión de las direcciones futuras sobre esta temática.

Tras el postproceso de cadenas puede aparecer la necesidad de discernir entre las cadenas correctamente postprocesadas de aquellas que no lo son. Para dar solución a este tipo de problema se suele recurrir al establecimiento de un umbral de rechazo sobre una medida relacionada con la fiabilidad que se tiene a nivel de cadena, tras el postproceso automático, con el fin de distinguir las correctamente transcritas de las incorrectas. La optimización de umbrales de rechazo se ha estudiado ampliamente en diversidad de campos como pueden ser la estadística y el aprendizaje automático entre otros. A lo largo de la historia se han desarrollado aproximaciones, tanto genéricas como dependientes de la tarea, con objeto de minimizar, o al menos intentar controlar, los errores relacionados tanto con los falsos positivos como con los falsos negativos. El problema de la optimización del umbral surge originalmente en teoría de detección de la señal y particularmente en el reconocimiento de formas, siendo de aplicación en una gran variedad de aplicaciones científicas, como por ejemplo, sistemas biométricos y de diagnosis Swets [1988]. Estas aplicaciones suelen estar seriamente comprometidas con los diferentes riesgos relativos a los errores procedentes de falsos positivos y falsos negativos mencionados anteriormente (un ejemplo claro son las aplicaciones relacionadas con la diagnosis médica). En este tipo de sistemas el establecimiento del umbral de rechazo juega un papel relevante que ha sido ampliamente estudiado en la literatura.

En el campo de la detección de la señal, se han realizado extensos trabajos durante décadas entre los que cabe citar, Ozturk et al. [1996]. En este trabajo se utiliza la distribución generalizada de Pareto para aproximar las colas de la distribución de las medidas de un radar, y se propone el método de mínimos cuadrados de muestras ordenadas (OSLS) para estimar el umbral. Recientemente, en Broadwater and Chellappa [2010], se propuso un algoritmo, usando la teoría de valores extremos a través del uso de la distribución de Pareto generalizada

y un test estadístico de Kolmogorov-Smirnov, y se incorporó una forma de mantener adaptativamente bajas tasas de falsos positivos minimizando las diferencias entre las asunciones del modelo y los datos reales. En Hansen [2000], se hace uso de la regresión para estimar el umbral en sistemas de sensores en los que frecuentemente se utiliza una gran cantidad de datos, y donde la detección del objetivo está seriamente afectada por los falsos positivos, por lo que se hace imprescindible el establecimiento del punto de control operacional del sistema.

En estadística, la tasa de error, tal y como se definirá en el apartado 5.2, viene referenciada como *False Discovery Rate (FDR)*, que no es más que la probabilidad de aceptar la hipótesis nula (H_0) cuando realmente lo correcto hubiese sido aceptar la hipótesis alternativa (H_1). Este tipo de error se conoce también como error de tipo II ($\beta = P(H_0|H_1)$). En Benjamini and Hochberg [1995], se establece un procedimiento para controlar la proporción esperada de hipótesis H_1 erróneamente rechazadas, en problemas de test significativos múltiples y aplicados a diagnosis médica. Para ello se establece el control del error de tipo II (β) en lugar de lo que comúnmente controlan los test de hipótesis, que es el error de tipo I ($\alpha = P(H_1|H_0)$).

En el ámbito del reconocimiento de formas, los trabajos iniciales de Chow [1970] describen el compromiso existente entre el error y el rechazo que aparece en los problemas de clasificación y que se representa gráficamente a través de la curva *Error-Reject*. En una curva de error frente a rechazo, se ofrece para cada valor de umbral, el error cometido dentro del conjunto de cadenas aceptadas y la cantidad correspondiente de cadenas rechazadas, tal y como se describe en Chow [1970], donde aparece una regla óptima de rechazo y se presenta una relación general entre la probabilidad de error y la probabilidad de rechazo, estableciéndose que la tasa de error puede ser directamente evaluada a partir de una función de rechazo y discutiéndose algunas implicaciones prácticas.

Alguno de los trabajos que tratan directamente con el problema de controlar los errores relacionados con falsos positivos y falsos negativos se ha centrado en la mejora de la curva ROC, de la cual se pueden encontrar infinidad de análisis e interpretaciones en la literatura Fawcett [2006]. Otro tipo de curva también relacionada con la estimación del error es la curva *Precision-Recall* Rijsbergen [1979], que también permite analizar la relación entre las tasas de error y el umbral, de aceptación y de rechazo, utilizado. El trabajo de Landgrebe et al. [2006] nos presenta las curvas de *Precision-Recall Operating Characteristic (P-ROC)*, que son una versión modificada de la curva ROC donde aparece un factor de sintonización para controlar el número de falsos positivos esperados, el cual es un dato de entrada. El método podría verse como un esfuerzo que trata de compensar las diferencias existentes entre la distribución de los datos de entrenamiento y la distribución de los datos de test.

La calidad de las medidas de confianza, junto con los criterios de rechazo, son elementos clave en los campos relativos al aprendizaje automático y el procesado del lenguaje natural (LNP), incluyéndose en este último campo tanto la traducción automática como el reconocimiento automático del habla Gandrabur et al. [2006]; Ueffing and Ney [2007]. Las aproximaciones provenientes de estos campos frecuentemente se integran en sistemas de reconocimiento OCR y de reconocimiento de escritura manuscrita. En varios trabajos relacionados directamente con el postproceso de tales tareas, se proponen mejoras en las estrategias de rechazo y medidas de confianza orientadas a obtener procedimientos eficaces para la mejora

de la fiabilidad del reconocimiento que ayuden a discernir entre aquello bien reconocido de lo que ha sido mal reconocido Bertolami et al. [2006]; He et al. [2009]; Pitrelli et al. [2006]; Schlapbach et al. [2008].

En corrección de texto, Kae et al. [2009] utilizan una técnica que permite identificar un conjunto de palabras correctas haciendo uso de la probabilidad de que cualquier palabra procedente de una salida OCR sea incorrecta, utilizando para ello un análisis de casos peores. Lindberg et al. [1998] presentan técnicas de decisión a priori en la estimación del umbral en tareas relacionadas con la verificación del locutor. En diagnosis médica, los esfuerzos han ido en la línea de encontrar el compromiso entre el error y el rechazo. Hanczar and Dougherty [2008] formalizan un problema de optimización para minimizar la tasa de rechazo bajo la restricción de que la tasa de error de un clasificador de dos clases tuviese un límite en la tasa de error objetivo, predefinida ésta previamente por el usuario. En este trabajo los umbrales para cada clase se seleccionan basándose en las densidades de probabilidad condicional de un conjunto de entrenamiento, seleccionándose un umbral diferente para cada una de las clases.

En el contexto de una aplicación real, el establecer un umbral de rechazo automático a partir de una tasa de error especificada previamente, aliviaría el problema de gestionar las medidas de confianza por parte del usuario. En este sentido, el trabajo de Li and Sethi [2006b] propone una aproximación para el diseño de clasificadores con el requerimiento del problema del control de la tasa de error en un problema de dos clases. Mientras que su método resuelve objetivos relacionados que podrían extrapolarse a nuestro objetivo en cierta forma, su implementación requiere del uso de dos umbrales y diferentes funciones a la hora de calcular el conjunto de observaciones rechazadas. Li and Sethi [2006a] extiende su propio trabajo haciendo uso de la metodología *Active Learning*. Por otra parte, Hanczar and Dougherty [2008] aplicó la propuesta de Li and Sethi [2006b] en la clasificación de datos sobre expresión de genes formalizando el compromiso entre el error y el rechazo como un problema de optimización para el caso concreto de clasificación de dos clases en la expresión genética, donde la función objetivo consiste en la minimización de la tasa de rechazo, y el límite superior de tasa de error objetivo asumido por cada clase está en las restricciones. Por otra parte, Serrano et al. [2014] presentan el problema en el contexto de la supervisión del error en tareas de reconocimiento automático manuscrito interactivo y predictivo. El propósito de éste consiste en asistir al usuario en la localización de errores en la transcripción: los usuarios deciden sobre un umbral máximo de tolerancia del error asumible durante el reconocimiento (después de la supervisión), y el sistema ajusta de manera automática el esfuerzo de supervisión requerido basándose en la estimación de este error. En este caso, para el elemento que está siendo tratado en un momento dado, la estimación del error se basa en los elementos previamente supervisados, haciendo uso para ello de una estrategia similar a la propuesta en esta tesis, no siendo directamente aplicable a una muestra completa, particularmente cuando no se dispone de información supervisada.

Si lo que se pretende es mejorar la interacción persona-máquina, tanto en cadenas previamente postprocesadas como en situaciones de entrada directa de datos, con objeto de reducir el esfuerzo requerido a la hora de realizar dicha interacción, será necesario disponer de un método capaz de combinar las diferentes evidencias de entradas multimodales de las que se pueda disponer Bastide et al. [2004]; Müller and Weinberg [2011].

En Suhm et al. [1999] se presenta un modelo que es un primer paso hacia la formalización de la interacción multimodal para la corrección de errores en interfaces de automático del habla. Los resultados obtenidos en este trabajo muestran que las predicciones del modelo multimodal complementan y ayudan en la toma de decisiones referentes a la elección de la transcripción correcta.

En cuanto a la predicción de texto en Garay-Vitoria and Abascal [2006] se presenta un estudio sobre diferentes técnicas utilizadas en este campo. Las diferentes técnicas de predicción de texto tienen diferentes rendimientos, según el nivel de flexión que tiene el lenguaje en el que se introduce el texto. En Garay-Vitoria and Abascal [2010] se hace un estudio comparativo sobre el rendimiento de las diferentes técnicas a la hora de introducir el texto, en lenguajes con baja y con elevada flexión.

Por otra parte, en Toselli et al. [2010] se muestra un sistema multimodal e interactivo de reconocimiento de texto en documentos antiguos. Dicho sistema de reconocimiento manuscrito está basado en modelos HMM y modelos de lenguaje, y en él se hace un reconocimiento previo del texto reconocido. Tras este reconocimiento los errores son corregidos mediante la interacción del propio usuario con diferentes sistemas multimodales como pantallas táctiles, teclado y ratón. La corrección se realiza aceptando como válido el prefijo existente hasta la zona seleccionada sobre la que el usuario introduce una corrección, tras esto el sistema fusiona tanto la información anterior como la nueva información relativa a la interacción y propone una nuevo reconocimiento del sufijo que maximiza la probabilidad de todas las evidencias presentes en ese momento, el resultado consiste en una reducción del esfuerzo necesario para realizar dicha transcripción. En Alabau et al. [2014] se presenta un sistema iterativo multimodal para la transcripción de documentos históricos manuscritos. En dicho sistema se fusiona tanto la información procedente del texto manuscrito como la información procedente de un sistema de reconocimiento de la voz, a la hora de realizar las transcripciones por parte de un humano, observándose una mejora con el uso de la multimodalidad respecto a un sistema unimodal.

CAPÍTULO 2

MODELO MATEMÁTICO DE LOS AUTÓMATAS DE ESTADOS FINITOS CON PESOS

Índice del capítulo

Los transductores con pesos se utilizan en muchas aplicaciones relacionadas con el procesado de texto de habla o de imágenes Mohri [1997] Fernando et al. [1996] Culik and Kari [1997]. Se trata de autómatas en los cuales, cada transición es etiquetada con la habitual etiqueta de entrada pero además poseen una etiqueta de salida y un peso asociado a la transición. Los transductores se utilizan para establecer asociaciones entre dos tipos diferentes de fuentes de información. Los pesos se usan para modelizar la incertidumbre o la variabilidad en tales fuentes de información. Dichos transductores se pueden utilizar por ejemplo para asignar diferentes pronunciaciones a una misma palabra pero con diferentes probabilidades para cada pronunciación. La estimación de dichos pesos se puede llevar a cabo mediante el uso de conjuntos de datos y técnicas de aprendizaje estadístico. En este capítulo se va a establecer la base matemática de los WFST junto con la definición de ciertas operaciones que se pueden realizar sobre los FST y que son de gran interés para los casos prácticos que se expondrán en este trabajo en los siguientes capítulos.

2.1. Estructura algebraica

En este apartado se van a introducir las definiciones y notaciones utilizadas Mohri [2004].

Definición 1 *Una estructura algebraica es una n-tupla* $(a_1, a_2, \ldots, a_n)$*, en la que* a_1 *es un conjunto dado no vacío, y* $a_2, \ldots, a_n$ *consiste en un conjunto de operaciones que pueden ser aplicadas a todos los elementos de dicho conjunto* a_1*.*

Definición 2 *Una ley de composición interna es la operación* $\circ$*, definida sobre la estructura algebraica, en la que los elementos que intervienen en todo momento pertenecen al conjunto* A *de la estructura algebraica. (*$f : A \times A \times A \times \cdots \times A \to A$*)*

Definición 3 *Un semigrupo es una estructura algebraica de la forma* $(A, \circ)$ *donde* A *es un conjunto donde se ha definido una ley de composición interna binaria* $\circ$*. Un semigrupo cumple las siguientes propiedades:*

1. ***Operación interna:*** *Para cualesquiera dos elementos del conjunto* A *operados bajo* $\circ$*, el resultado pertenece al mismo semigrupo* A*. Es decir:* $\forall x, y \in A : x \circ y \in A$
2. ***Asociatividad:*** *Para cualesquiera elementos del conjunto* A *no importa el orden en el que se operen las parejas de elementos, mientras no se cambie el orden de los elementos, siempre dará el mismo resultado. Es decir:* $\forall x, y, z \in A : x \circ (y \circ z) = (x \circ y) \circ z$

Si además se cumple la propiedad ***conmutativa*** *que dice que un conjunto* A *tiene la propiedad conmutativa respecto a la operación interna* $\circ$ *si:* $\forall a, b \in A : a \circ b = b \circ a$*. Se dice que es un semigrupo conmutativo o abeliano.*

Definición 4 *Un monoide es un semigrupo con elemento neutro (para todo elemento* x *que pertenezca al conjunto* A*, existe un único elemento* e *de* A*, que cumple:* $\forall x \in A : \exists! e : e \circ x = x \circ e = x$*). Si además se cumple la propiedad* ***conmutativa*** *se dice que el monoide es* ***conmutativo*** *o* ***abeliano****.*

Definición 5 (Kuich and Salomaa [1986]) *Un sistema* $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ *es un semianillo si:* $(\mathbb{K}, \oplus, \bar{0})$ *es un monoide conmutativo con elemento identidad* $\bar{0}$*;* $(\mathbb{K}, \otimes, \bar{1})$ *es un monoide con elemento identidad* $\bar{1}$*;* $\otimes$ *se distribuye sobre* $\oplus$*; y* $\bar{0}$ *es un aniquilador para* $\otimes$*: para todo* $a \in \mathbb{K}, a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$.

Así pues, un semianillo es una anillo que puede carecer de negación, es decir puede carecer de elemento simétrico para el operador $\oplus$. En la tabla 2.1 se citan algunos de los semianillos más familiares.

SEMIANILLO	CONJUNTO	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$
Booleano	$\{0, 1\}$	$\vee$	$\wedge$	0	1
Probabilidad	$\mathbb{R}_+$	$+$	$\times$	0	1
Log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{log}$	$+$	$+\infty$	0
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0

Tabla 2.1: Ejemplos de semianillos. $x \oplus_{log} y = -\log(e^{-x} + e^{-y})$

Se dice que un semianillo es *conmutativo* cuando la operación multiplicativa $\otimes$ es *conmutativa*. Se dice que es *divisible por la izquierda* si para cualquier $x \neq \bar{0}$, existe $y \in \mathbb{K}$ tal que $y \otimes x = \bar{1}$, es decir todos los elementos de $\mathbb{K}$ admiten una inversa por la izquierda. Se dice que un sistema $((K), \oplus, \otimes, \bar{0}, \bar{1})$ es débilmente divisible por la izquierda si para cada x e y en $\mathbb{K}$ tal que $x \oplus y \neq \bar{0}$, existe al menos un z tal que $x = (x \oplus y) \otimes z$. Cuando la operación $\otimes$ es cancelativa, z es único y se puede escribir: $z = (x \oplus y)^{-1} x$. Cuando resulta que z no es un elemento único, se asume que es posible tener un algoritmo para encontrar una de las posibles soluciones z y llamarla $(x \oplus y)^{-1} x$. Además, se asume que z puede ser encontrado de manera *consistente*, lo que quiere decir que: $((u \otimes x) \oplus (u \otimes y))^{-1}(u \otimes x) = (x \oplus y)^{-1} x$ para cualquier $x, y, u \in \mathbb{K}$ tal que $u \neq \bar{0}$. Un semianillo es *cónico* o *positivo* si para cualquier x e y en $\mathbb{K}, x \oplus y = \bar{0}$ implica $x = y = \bar{0}$. En las siguientes definiciones, se asume que $\mathbb{K}$ será un semianillo por la izquierda o semianillo.

Definición 6 *Un transductor de estados finitos con pesos* T *sobre un semianillo* $\mathbb{K}$ *es una 8-tupla* $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ *donde:* Σ *es el alfabeto finito de entrada al transductor;* Δ *es el alfabeto de salida finito;* Q *es el conjunto finito de estados;* $I \subseteq Q$ *el conjunto de estados iniciales;* $F \subseteq Q$ *es el conjunto de estados finales;* $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \times \mathbb{K} \times Q$ *un conjunto finito de transiciones;* $\lambda : I \to \mathbb{K}$ *la función de asociación de pesos inicial de* F *a* $\mathbb{K}$ *y* $\rho : F \to \mathbb{K}$ *la función de asociación de pesos final de* F *a* $\mathbb{K}$.

Se denota por $|T|$ a la suma del número de estados y transiciones de T. Los *autómatas con pesos* se definen de manera similar pero omitiendo simplemente la entrada o la salida.
Dada una transición $e \in E$, se denota por $p[e]$ el origen o estado previo y $n[e]$ el estado destino o nuevo estado, y $w[e]$ su peso. Un camino $\pi = e_1 \dots e_k$ es un elemento de E^* con transiciones consecutivas: $n[e_{i-1}] = p[e_i], i = 2, \dots, k$. Se extiende n y p a caminos poniendo: $n[\pi] = n[e_k]$ y $p[\pi] = p[e_1]$. La función de peso w puede ser extendida a caminos definiendo el peso de un camino como el $\otimes$-producto de los pesos de sus transiciones constituyentes: $w[\pi] = w[e_1] \otimes \cdots \otimes w[e_k]$. Se denota por $P(q, q')$ al conjunto de caminos desde q a q' y por $P(q, x, y, q')$ al conjunto de caminos desde q a q' con etiqueta de entrada $x \in \Sigma^*$ y etiqueta de salida $y \in \Delta^*$. Estas definiciones pueden ser extendidas a subconjuntos

$R, R' \subseteq Q$, por: $P(R, x, y, R') = \cup_{q \in R, q' \in R'} P(q, x, y, q')$. Un transductor T es regulado si el peso de salida asociado por T a cualquier par de cadenas de entrada/salida (x, y) por:

$$[[T]](x, y) = \bigoplus_{\pi \in P(I,x,y,F)} \lambda[p[\pi]] \otimes w[\pi] \otimes \rho[n[\pi]] \tag{2.1}$$

está bien definido y en $\mathbb{K}$. $[[T]](x, y) = \bar{0}$ cuando $P(I, x, y, F) = \emptyset$. Si para todo $q \in Q$ $\bigoplus_{\pi \in P(q,\epsilon,\epsilon,q)} w[\pi] \in \mathbb{K}$ entonces T está regulado. En particular, cuando T no tiene ningún ϵ-ciclo, se dice que está regulado. Se define el domino de T, $Dom(T)$, como: $Dom(T) = \{(x, y) : [[T]](x, y) \neq \bar{0}\}$.

2.2. Composición de autómatas transductores

La operación de composición es un algoritmo fundamental a la hora de crear transductores con pesos a partir de otros más sencillos. Sea $\mathbb{K}$ un semianillo conmutativo y sean T_1 y T_2 dos transductores con pesos definidos sobre $\mathbb{K}$ de tal manera que el alfabeto de entrada del transductor T_2 coincide con el alfabeto de salida del transductor T_1 y la suma $\bigoplus_z T_1(x, z) \otimes T_2(z, y)$ está bien definida en $\mathbb{K}$ para todo $(x, y) \in \Sigma^* \times \Omega^*$. Entonces, el resultado de la composición de T_1 y T_2 es un nuevo transductor con pesos que se denota como $T_1 \circ T_2$ y que está definido para todo x, y:

$$[[T_1 \circ T_2]](x, y) = \bigoplus_z T_1(x, z) \otimes T_2(z, y) \tag{2.2}$$

Los estados en la composición, $T_1 \circ T_2$, de dos autómatas transductores con pesos, T_1 y T_2, se identifican con pares de estados que proceden de un estado de T_1 y un estado de T_2. La siguiente regla especifica el cálculo de una transición de $T_1 \circ T_2$ a partir de las transiciones apropiadas de T_1 y T_2:

$$(q_1, a, b, w_1, q_2)\, y\, (q_1', b, c, w_2, q_2') \rightarrow ((q_1, q_1'), a, c, w_1 \otimes w_2, (q_2, q_2')) \tag{2.3}$$

En el peor de los casos, todas las transiciones del transductor T_1 que abandonan un estado q_1 podrían encajar con todas las transiciones del transductor T_2 que abandonan el estado q_1', así pues la complejidad espacial y temporal de la operación de composición es cuadrática: $O(|T_1|\,|T_2|)$. Sin embargo, y a pesar de este inconveniente con los costes es posible retrasar la operación de composición para poder realizarla al vuelo y sólo para aquellas partes de los transductores que se requieran cuando sea necesario.

La intersección de un autómata con pesos y la composición de de un transductor de estados finitos son casos especiales de la composición de transductores con pesos. La intersección se corresponde con el caso en el que las etiquetas de entrada y salida coinciden y la composición de transductores sin pesos se obtienen simplemente mediante la omisión de tales pesos.

2.3. Características de los transductores de estados finitos con pesos

2.3.1. Definición

Tal y como queda patente en Mohri et al. [2000]; Vidal et al. [2005], se puede ver a un *autómata transductor de estados finitos con pesos (WFST)* como una generalización de un *autómata de estados finitos (FSA)*. A su vez, un FSA se puede representar como un grafo dirigido, en el que los nodos representan a estados y los arcos representan a las transiciones entre estados. En dicho grafo, cada transición posee una etiqueta con un símbolo procedente de un alfabeto Σ. De manera formal se define a un FSA como una 5-tupla $(Q, q_o, F, \Sigma, \delta)$ donde Q es un conjunto de estados finito, $q_0 \in Q$ representa al estado inicial, $F \subseteq Q$ es un conjunto de estados finales, Σ es un conjunto finito de símbolos y $\delta : Q \times \Sigma \rightarrow Q$ es el conjunto de transiciones entre estados. Cada transición, $t \in \delta$, se etiqueta con un símbolo $s(t) \in \Sigma$. Los FSA se usan para aceptar o rechazar conjuntos de cadenas definidas sobre el alfabeto Σ. Dada una cadena $w \in \Sigma^*$, se aceptará w, sólo si hay un camino desde el estado inicial al estado final, en el grafo FSA definido, cuyas etiquetas en las transiciones forman la cadena w cuando éstas son concatenadas.

Por otra parte, en un *autómata transductor de estados finitos (FST)*, aparece un nuevo alfabeto de salida, Δ, y además sobre cada transición aparece una etiqueta con un símbolo de entrada, e_j, donde $e_j \in \Sigma$ y un símbolo de salida, s_j, donde $s_j \in \Delta$. Por lo tanto, la función δ, se define ahora, como $\delta : Q \times \Sigma \rightarrow Q \times \Delta$. La utilidad de los FST es la de *transducir* cadenas de un lenguaje de entrada definido sobre el alfabeto Σ, en cadenas de un lenguaje de salida definido sobre el alfabeto Δ. La versión con pesos de un FST denominada como WFST incluye además un valor numérico o peso en cada una de las transiciones. Dicho valor puede usarse para diferentes fines, como por ejemplo, representar el coste de elección de un determinado camino. Además, cada estado q tiene un peso inicial $w_i(q)$ y un valor final $w_f(q)$. Así pues, un estado q en un WFST es *i*nicial si $w_i(q) \neq \bar{0}$ y *f*inal si $w_f(q) \neq \bar{0}$.

En un WFST, es posible obtener una cadena de salida más corta que la cadena de entrada si Δ incluye al símbolo vacío ε y también se pueden obtener cadenas más largas si ε está incluido además en Σ. En el caso en el que tanto Σ como Δ admitan al símbolo ε, se diferenciarán entre tres tipos de operaciones de transición en un FST: *sustitución*, cuando un símbolo de entrada e_j es transducido a un símbolo de salida s_j donde $e_j, s_j \neq \varepsilon$ y no necesariamente ocurre que $e_j \neq s_j$; *inserción*, cuando ε es transducido a un símbolo de salida s_j; y *borrado*, cuando un símbolo de entrada e_j es transducido a ε.

Existe un tipo especial de transductores FST o WFST, que se conocen como *transductores identidad*, en el que para cada una de las transiciones el símbolo de entrada coincide con su símbolo de salida. Este tipo de transductor tiene una equivalencia directa con un FSA o WFSA en su versión con pesos.

2.3.2. Propiedades de la composición de los autómatas conjuntos y los autómatas condicionales

Dado un transductor cuyos pesos representan probabilidades, si todos los arcos de salida de un estado dado, junto con el peso de estado final, tienen una probabilidad total igual a la unidad, entonces las relaciones de (e_j, s_j) que este transductor produce se conoce como relaciones conjuntas ($\cap$) y tienen una probabilidad $P(e_j, s_j)$. Por el contrario, si los arcos de salida de un estado dado y un símbolo de entrada dado tienen una probabilidad total igual a la unidad, entonces la relación se conoce como condicional (|) y por lo tanto se representa a la probabilidad $P(s_j|e_j)$ Eisner [2002].

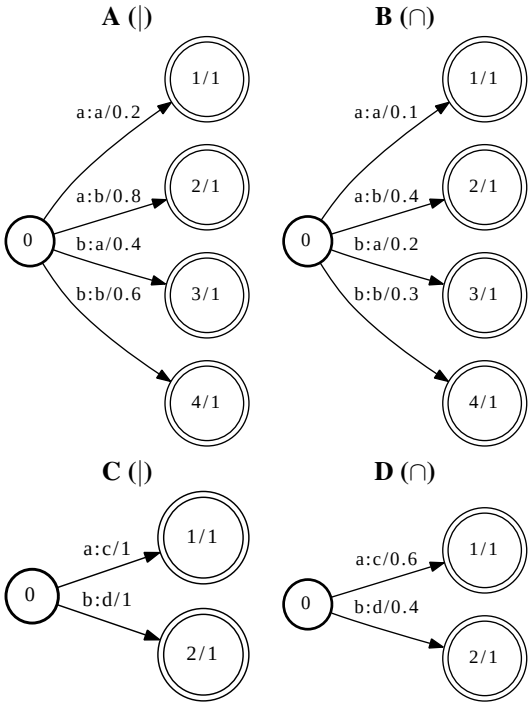

Figura 2.1: En A y C se presentan dos autómatas modelados según la ley de la probabilidad condicional y en B y D se pueden observar dos autómatas modelados según la ley de la probabilidad conjunta. En A y B el alfabeto de entrada y salida es el mismo, sin embargo C y D presentan alfabetos de entrada y salida diferentes.

En la figura 2.1 se muestran cuatro autómatas modelados de manera diferente según la ley de probabilidad conjunta (B,D) y condicional (A,C) y sobre alfabetos de salida que pueden coincidir con el de entrada (A,B) o no (C,D). Estos autómatas conformarán nuestra base para poder observar diferentes propiedades de la operación de composición según las características y el orden de los autómatas que se compongan.

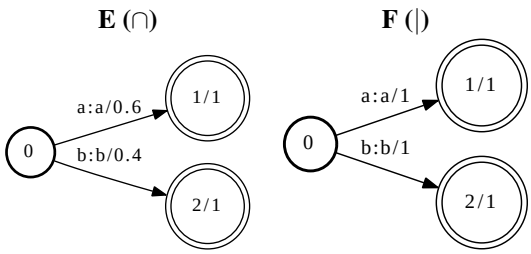

Figura 2.2: Se muestran dos autómatas transductores identidad modelados según la ley de probabilidad conjunta (E) y según la ley de probabilidad condicional (F).

En la figura 2.2 se observan dos autómatas transductores identidad. En este tipo especial de autómatas transductores la modelización bajo la ley de la probabilidad conjunta ofrece información probabilística en cada una de las transiciones, sin embargo, si se modelan desde el punto de vista de la ley de probabilidad condicional entonces todas las aristas son equiprobables con valor igual a uno.

En la figura 2.3 se observan las posibles combinaciones de composición de autómatas transductores desde la perspectiva de la ley de probabilidad conjunta y condicional. A raíz de tales composiciones se puede constatar el resultado probabilístico de la operación de composición:

$\cap \circ |$**:** En C1 se observa que el resultado de este tipo de composición da lugar a un autómata estocástico conjunto.

$\cap \circ \cap$**:** En C2 se observa que el resultado de este tipo de composición resulta en un autómata no estocástico.

$| \circ |$**:** En C3 se observa que el resultado de una composición de dos autómatas condicionales resulta en un nuevo autómata también condicional.

$| \circ \cap$**:** En C4 se observa que una composición de un autómata condicional por uno conjunto resulta en un autómata no estocástico.

A la vista de estos resultados se puede concluir que para que el autómata transductor resultante de una cadena de composiciones sea también estocástico desde el punto de vista de la ley de la probabilidad conjunta, entonces la composición de estos se ha de realizar según el orden de la ley de la probabilidad total, en la que si se tienen tres sucesos (A,B,C) entonces la probabilidad del suceso ABC se calcula, según la regla de la cadena, como: $P(ABC) = P(A)P(B|A)P(C|AB)$, donde $P(A)$ vendría a simbolizar nuestro autómata conjunto y el resto los diferentes autómatas condicionales.

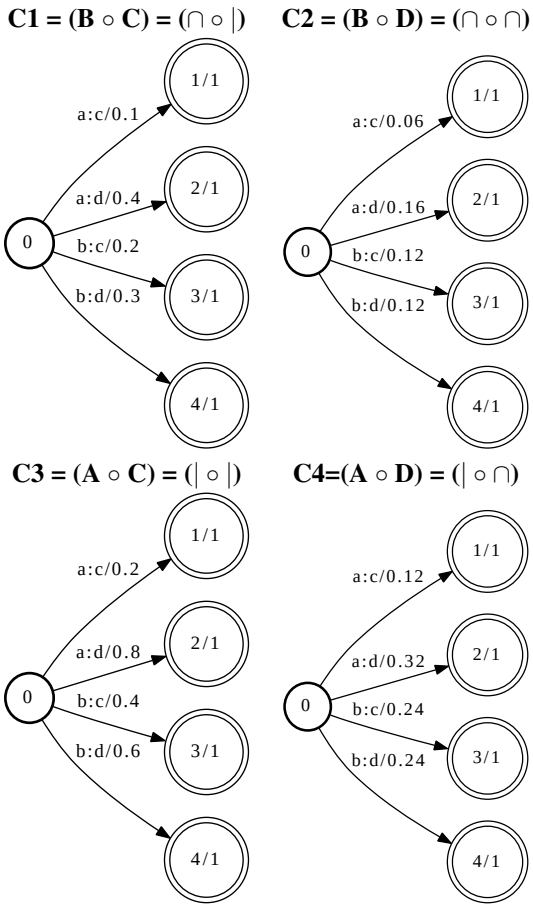

Figura 2.3: Se muestra el resultado de la composición de diferentes autómatas procedentes de la figura 2.1. En C1 se muestra el resultado de la composición de un autómata conjunto (B) con otro condicional (C). En C2 tenemos la composición de los autómatas B y D que son conjuntos. El autómata C3 muestra la composición de los autómatas A y C que son condicionales y finalmente el autómata C4 muestra la composición de un autómata condicional (A) con un autómata conjunto (D).

CAPÍTULO 3

METODOLOGÍA

Índice del capítulo

3.1. El sistema general propuesto

Se propone un sistema capaz de combinar varias fuentes de información, incluyendo las evidencias probabilísticas, para tratar con el problema de procesar cadenas de hipótesis iniciales que vienen de diferentes subsistemas de entrada. La finalidad es obtener una salida mejorada de acuerdo a las restricciones impuestas por distintos modelos.

A continuación se enumeran las fuentes de información habituales, a falta de otras que se irán añadiendo conforme a la información disponible en cada momento para la resolución de un problema dado:

1. La *hipótesis de entrada*, que puede ser tan sencilla como una secuencia de símbolos o tan compleja como un grafo que represente un conjunto de frases probabilísticas que pertenecen a una gramática o lenguaje. Esto es lo que denominaremos *modelo de hipótesis (HM)*. Esta es la información inicial, que puede venir de la salida de un clasificador de *reconocimiento óptico de caracteres (OCR)*, un sistema de *reconocimiento automático del habla (SR)*, o cualquier otro sistema de entrada general.

2. El *modelo de errores* esperados en la hipótesis de entrada, esto es lo que denominaremos como *modelo de error (EM)*. En este tipo de modelo se ofrece información probabilística sobre cuáles son los errores más habituales. Esta cuantificación servirá para que el sistema sea capaz de ponderar el coste entre las diferentes transducciones pues los errores más habituales serán menos costosos (más probables) que los poco habituales (menos probables).

3. El lenguaje al que se espera que pertenezcan el conjunto de cadenas resultantes, esto es lo que llamaremos *modelo de restricciones (CM)* o *modelo de Lenguaje* y representará a las cadenas de salida válidas.

Al combinar de la manera apropiada todos los modelos, el sistema propondrá, ante cualquier entrada, la cadena más probable compatible con el modelo CM. Para ello lo que hará es buscar la transformación más probable de una cadena de un modelo HM a una cadena que pertenezca a CM a través de las operaciones de error que están definidas en EM. Aunque aquí se explica la técnica como una transformación desde HM a CM, este método puede ser visto como una aproximación a un canal con ruido, donde la cadena observada se considera una versión ruidosa de la verdadera cadena Brown et al. [1993]. Así, CM genera una cadena libre de errores con una determinada probabilidad y EM (modelo de ruido) decide, si es necesario o no, insertar errores para producir la cadena observada en HM, Park and Levy [2011].

Las aproximaciones que se van a ir proponiendo en los diferentes problemas que se irán planteando se van a resolver mediante un método flexible y genérico, porque HM no está pensado para codificar únicamente una cadena, sino que es capaz de codificar entradas más complejas, conforme se puede observar en el apartado 3.2.2.

Puede ocurrir también que la cadena propuesta por el sistema tras la transformación no sea la pretendida. En este caso, se permitiría la interacción con el usuario para corregir el

error, lo que nos lleva a una nueva fuente de información que vamos a modelar a través del *modelo de interacción (IM)* conforme se muestra en el apartado 3.2.4. Este nuevo modelo se puede combinar dinámicamente a la vez que el usuario va interactuando con el sistema, con el resto de modelos estáticos comentados anteriormente. El objetivo en este caso no es otro que el obtener la cadena correcta realizando un mínimo número de interacciones, por parte del usuario, e incrementando así la productividad en la transcripción de cadenas.

Se puede incluso modelar el hecho de que el usuario cometa errores mientras interactúa con el sistema (por ejemplo, pulsando una tecla cercana a la deseada, pulsando dos veces la misma tecla, etc.). En este caso, se deberá de asociar al modelo de interacción con el usuario su propio modelo de error que permita la recuperación de los errores introducidos durante el proceso de entrada, por parte del usuario, y que será independiente del modelo de error general descrito inicialmente. De hecho, el modelo IM puede verse como un segundo modelo de hipótesis, que conduce a un sistema multimodal tolerante a fallos e interactivo en el que se combinan varias entradas para proponer una salida. El sistema así propuesto, tiene un doble propósito: el primer propósito consiste en transformar la evidencia observada, que queda reflejada en HM, en una cadena válida perteneciente a CM, basándose en una aproximación donde se busca la maximización de la probabilidad, o de manera equivalente, la minimización del coste de transformación, junto a un segundo propósito, en el caso en el que la cadena propuesta no sea la pretendida. Este segundo propósito consiste en conseguir la cadena correcta final con el mínimo número de interacciones por parte del usuario.

Se propone representar cada uno de los modelos presentados (HM, EM, CM, IM) por medio de un transductor de estados finitos con pesos (WFST) y componer todos ellos de la manera adecuada con el objetivo de dirigir el problema de búsqueda de la cadena más probable en CM de acuerdo a la hipótesis (HM), los posibles errores (EM) y la entrada por parte del usuario (IM). Este problema se resuelve de manera sencilla, mediante la búsqueda de la cadena más probable en el transductor resultado de la composición de autómatas WFST tal y como se comenta en 3.2.5.

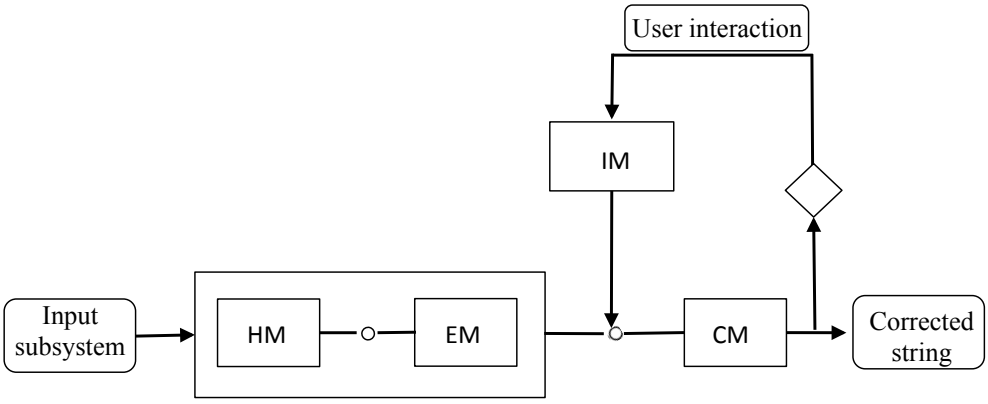

Figura 3.1: Esquema de postproceso interactivo multimodal.

La figura 3.1 muestra un sistema general de interacción multimodal, donde el operador $\circ$ representa la composición de transductores. En un primer paso, HM, EM y CM son compuestos, lo que permite transducir cualquier cadena $x \in L_{HM}$ en cualquier otra cadena $y \in L_{CM}$, donde L_{HM} representa al lenguaje generado por el autómata HM y L_{CM} sería el lenguaje generado por el autómata CM, de acuerdo a las diferentes operaciones de error previamente definidas en EM (asumiendo que EM transduce cualquier cadena en cualquier otra). En la fase de decodificación, se busca la más probable de estas transducciones, o equivalentemente, aquella cadena cuya transformación a partir de la entrada ha resultado menos costosa. Si la salida propuesta no es la que el usuario desea, entonces se permite la interacción dinámica para mejorar la cadena resultante. En este caso, se crea un WFST dinámico adicional al que hemos llamado IM y se compone con HM$\circ$EM, lo que impondrá nuevas restricciones al sistema. En la dirección que aparece en el pie de página se puede acceder a un demostrador web del esquema anteriormente planteado[a].

En esta aproximación propuesta, no se hace ninguna asunción sobre el lugar de procedencia de la hipótesis inicial, o sobre cómo el usuario puede interactuar con el sistema. La hipótesis inicial podría ser la salida de una Interfaz persona-máquina como OCR, entrada táctil, sistema de reconocimiento de gestos, etc., sensores de entrada en un proceso físico, una cadena biológica como ADN o cadenas de proteínas o incluso una combinación de varios de ellos. La interacción con el usuario se puede hacer típicamente mediante un medio físico, basado en pantallas táctiles, o teclado reducido (como en un teléfono móvil), pero también por medio de voz o basados en reconocimiento de gestos, sistemas de reconocimiento OCR On-line, etc. Como ejemplo de ello, se puede ver el capítulo 6 del trabajo que aquí se presenta.

3.2. Transductores de estados finitos

Los FST y WFST se consideran especialmente flexibles y potentes debido a algunas de sus propiedades fundamentales. Concretamente, nos vamos a centrar en la operación de *composición* Riley et al. [1997]. Dados dos transductores T_1 y T_2, si T_1 transduce la cadena $x \in \Sigma$ a la cadena $y \in \Delta$ con peso w_1 y T_2 transduce $y \in \Delta$ a $z \in \Gamma$ con peso w_2, entonces su composición $T_3 = T_1 \circ T_2$, es un nuevo transductor que transduce x a z con peso $w_1 \otimes w_2$, véase la tabla 2.1. En este ejemplo, la operación de composición calcularía la intersección de T_1 y T_2 sobre el alfabeto Δ y construiría el transductor T_3 que tendrá definidos los símbolos de entrada sobre Σ y los símbolos de salida sobre Γ.

A continuación, se van a exponer las características generales de los principales autómatas que se usarán en las diferentes aplicaciones presentadas en el marco de este trabajo. El objetivo aquí, no es otro que el de mostrar el potencial y la flexibilidad que los WFST ofrecen en la resolución de diversas tareas. Las aplicaciones que aquí se presentan están orientadas, sobre todo, a la corrección automática y ayuda en la corrección de cadenas. Éstos autómatas pueden verse como pequeñas piezas independientes que modelizan un tipo de información muy concreta. En función del grado de información que se quiera aportar, y dependiendo de lo que se pretenda finalmente en cada una de las tareas, se unirán algunas o todas las piezas en

[a] https://demos.iti.upv.es/hi/

un orden adecuado. Esta unión se realizará, como si de un puzzle se tratara, mediante el uso de la operación de composición de autómatas. Esto conformará un todo, en forma de autómata, que unifica todas y cada una de las partes informativas, anteriormente creadas de manera independiente. El objetivo de todo esto, consiste en dar solución a los diferentes problemas planteados en las diferentes aplicaciones que aquí se presentan, representando de manera ponderada, todas las soluciones compatibles con el problema en forma de autómata. A partir de esta representación la solución final se encontrará mediante una búsqueda del mejor camino que se pueda encontrar sobre el autómata resultado. Los cuatro autómatas básicos que se van a plantear son: el modelo de restricciones (CM) que establece las cadenas finalmente admisibles por el sistema; el modelo de hipótesis (HM) que modela la información inicial de la que se dispone; el modelo de error (EM) que va a ofrecer información cuantificada relativa a los tipos de errores que se pueden dar; el modelo de interacción con el usuario (IM) que modela las entradas introducidas por el propio usuario en un momento dado. Posteriormente, se presentará un problema multimodal al que se le añade un nuevo modelo de hipótesis, junto a su propio modelo de error para ejemplificar la facilidad en la fusión de diferentes informaciones mediante esta tecnología.

3.2.1. El modelo de restricciones (CM)

El objetivo final del modelo de restricciones persigue la codificación de las restricciones impuestas por el sistema. En el tipo de tareas que se van a plantear aquí estas restricciones serán las cadenas admitidas como válidos por el sistema final. Este autómata representará al modelo de lenguaje finalmente aceptado por el sistema. Entre los diferentes niveles en los que un lenguaje puede ser modelado, el más bajo de ellos es el nivel de palabra. Para establecer las restricciones léxicas a nivel de palabra se realizará el establecimiento de relaciones entre los caracteres que forman parte de dichas palabras. El modelo de lenguaje puede ser cerrado, con lo que sólo se admitirían las palabras que se han modelado en el propio autómata, o abierto, en este caso el autómata tendría parte de las estructuras de las palabras y expresiones admisibles, siendo posible el generar nuevas palabras o expresiones, a partir de semiestructuras de las ya vistas. El transductor final que modela a las restricciones será un transductor identidad, pues los símbolos de entrada y salida en cada una de las transiciones serán el mismo.

Para la creación del modelo CM se propone un algoritmo de inferencia gramatical que permite construir una máquina de estados finitos que acepta el Modelo de Lenguaje k-Testable más pequeño en sentido estricto Garcia and Vidal [1990] consistente con una muestra de restricciones representativa de la tarea. El conjunto de cadenas aceptadas por dicho autómata es equivalente a un modelo de lenguaje clásico obtenido haciendo uso de n-gramas, para $n = k$. La extensión estocástica de un modelo k-Testable básico se realiza a través de la estimación de la probabilidad mediante la frecuencia de las reglas asociadas a la gramática, evaluándolas de acuerdo a su frecuencia de utilización por las cadenas de entrada. Este cálculo se lleva a cabo de manera incremental y simultánea durante el proceso de inferencia.

La principal ventaja del ajuste elegido reside en su flexibilidad. La muestra de restricciones puede ser tan simple como un simple léxico (lista de secuencia de símbolos, con cada posible secuencia apareciendo una única vez), una lista de secuencias extraídas de ejemplos

reales de la tarea (con cada una de ellas apareciendo tantas veces como así sea en la muestra real), una lista de secuencias con cadenas o categorías de cadenas como los símbolos de una gramática, etc. Únicamente en el primer caso, cuando se dispone de un léxico clásico, no se requiere que el autómata sea estocástico, puesto que un léxico no es una muestra representativa. En cualquier otro caso el modelo es capaz de aprovechar la ventaja de la información probabilística presente en los datos.

Se puede hacer uso del valor de k para controlar el comportamiento del modelo. En un modelo determinista (donde k es igual al tamaño de la longitud de la cadena más larga en la muestra) se obtiene un método de postproceso donde, únicamente las secuencias que existen en la muestra son válidas, pero si k es puesto a un valor más bajo, el resultado será un modelo de n-gramas clásico, donde las secuencias de salida pueden no estar en la muestra de referencia.

Es de destacar que la naturaleza estocástica o probabilística de la gramática subyacente, por ejemplo al tener en cuenta las frecuencias relativas de las diferentes secuencias de símbolos, no depende de la elección de k. Un modelo determinista se puede basar en una gramática estocástica, donde la base de datos de restricciones es una muestra real de secuencias y k es grande, y un modelo no determinista podría hacer uso de una gramática no probabilística, cuando la base de datos consiste en un simple léxico y k es pequeña.

La Figura 3.2 muestra un transductor probabilístico identidad asociado a la muestra S={aba, abb, ba, bac} y $k = 3$. En esta descripción, por conveniencia, hemos utilizado un transductor con los mismos símbolos de entrada y salida en cada transición. Este tipo de transductor se denomina transductor identidad, y puede verse como un aceptor o generador del lenguaje $L(S)$. Los pesos en las transiciones (probabilidades en este caso) se muestran en cada arco. Las probabilidades de los estados finales (estados con doble círculo) se muestran tras el número de estado.

Dada una secuencia s, $P(s)$ se calcula como el producto de las probabilidades a lo largo del camino (incluyendo la probabilidad de estado final) que acepta (o produce) s. Como el transductor propuesto representa relaciones de conjunción, $P(s)$ representa la probabilidad de que s sea aceptada (o producida) por el CM.

La complejidad asintótica temporal y espacial del procedimiento de construcción de un modelo de lenguaje podemos encontrarla en Garcia and Vidal [1990] y en nuestro caso es, O(kn log n) y O(n), respectivamente, donde n es el tamaño de la muestra del lenguaje. En la práctica el tiempo de construcción del modelo de lenguaje no es un problema pues se suele construir una sola vez, de manera previa a su uso.

3.2.2. El modelo de hipótesis (HM)

El modelo HM codifica la información inicial de la que se dispone y que será nuestra entrada. Este modelo consiste en un conjunto de hipótesis probabilísticas representadas mediante una máquina de estados finitos con pesos (WFST). De manera similar al modelo de restricciones (CM), el modelo de hipótesis (HM) es por lo general un modelo probabilístico

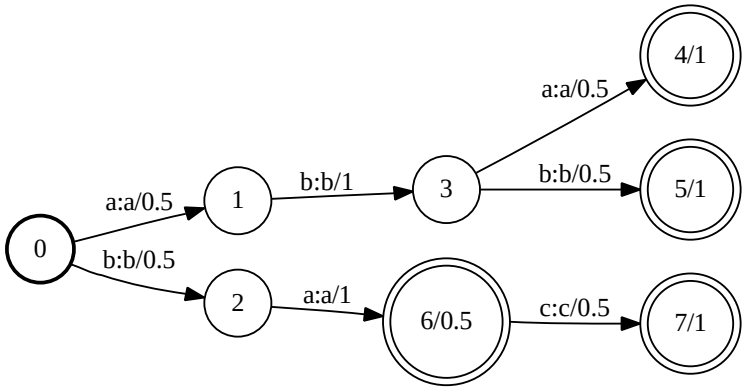

Figura 3.2: Ejemplo de un transductor identidad representando a un modelo de restricciones.

donde cada camino del modelo representa a una hipótesis h_i. La concatenación de las aristas desde un estado inicial a otro final (esto es, de un camino) forma una hipótesis, h_i, y el conjunto de todos los caminos posibles forma la hipótesis completa h. Dependiendo de la tarea que se esté modelizando la estructura del autómata WFST puede ser muy variable y así podemos encontrar saltos entre estados hacia delante o hacia atrás, ciclos, diferentes estados iniciales y finales, etc., dependiendo de lo que se considere factible en el problema en cuestión. El objetivo perseguido consiste en encontrar la transducción más probable de una cadena del modelo inicial HM en una cadena del modelo final CM.

Las hipótesis de entrada pueden contener errores debido a la incertidumbre, al error cometido por los clasificadores, a errores tipográficos etc. Así pues, si S_{HM} es el conjunto de todas las cadenas producidas por el modelo HM y S_{CM} el conjunto de todas las cadenas aceptadas por el modelo CM, entonces la composición de HM y CM define un nuevo WFST que acepta $S_{HM} \cap S_{CM}$. Sin embargo, el transductor compuesto no aceptaría ninguna cadena que no perteneciera a S_{HM}. Esta es una restricción muy fuerte debido al hecho de que por lo general, los sistemas no son procesos libres de errores, por lo tanto, se debería considerar el intercalar un modelo de error que sirva de puente entre el modelo HM y CM y que permita corregir las posibles transiciones erróneas tal y como se expone en el apartado 3.2.3.

Concretamente, y a modo de ejemplo, se ha tomado el transductor HM de una tarea OCR dedicada al reconocimiento de campos en formularios y que se expone en el capítulo 4. En este caso el modelo HM representa la salida del clasificador OCR. Suponiendo que nuestra cadena tiene una longitud de m caracteres y que el clasificador clasifica entre n categorías, la salida de dicho clasificador OCR se puede representar como un WFST identidad con $m + 1$ estados y n' transiciones entre cada par de estados que tienen probabilidad mayor que cero, donde $n' \leq n$.

La figura 3.3 muestra el ejemplo de un WFST con alfabeto $[a, b, c]$ que representa la entrada de símbolos [0.8, 0.2, 0.0],[0.1, 0.7, 0.2],[0.0, 0.6, 0.4]. Esto significa que el primer símbolo es a con probabilidad 0.8 o b con probabilidad 0.2, el segundo símbolo es a, b o c con probabilidades 0.1, 0.7 y 0.2 respectivamente, y así sucesivamente. Las transiciones

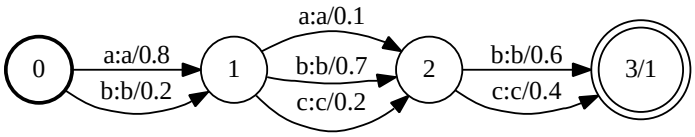

Figura 3.3: Ejemplo de un transductor identidad que representa un modelo de hipótesis con incertidumbre. Las probabilidades *a posteriori* de este subsistema de símbolos de entrada se muestran mediante pesos en los arcos. Dichos pesos representan el grado de fiabilidad que el clasificador tiene para cada símbolo en el instante que le corresponde y que viene determinado por los estados que se encuentran relacionados mediante aristas.

que tienen asignada una probabilidad de cero no se muestran en el grafo. Tal y como ha quedado patente en la anterior figura, en lugar de trabajar exclusivamente con la hipótesis más probable (abb), en este modelo transductor se modela además la incertidumbre del subsistema de entrada, dando a dicha incertidumbre el peso que le corresponde a través de un valor probabilístico ofrecido por el propio clasificador. Con ello se pretende reducir el número de cadenas finalmente erróneas.

3.2.3. El Modelo de Error (EM)

Se pueden dar casos en los que ninguna de las secuencias de símbolos de las hipótesis de entrada son posibles de acuerdo al modelo de restricciones, o puede darse el caso que una cadena que no pertenezca al conjunto de cadenas posibles de un modelo HM sea más probable que cualquiera de las opciones propuestas. En un modelo clásico, en el que el modelo de restricciones se implementa mediante el uso de $n-$gramas, este problema se evita mediante un procedimiento de *suavizado*. En nuestro caso, sin embargo, estas variaciones y sus probabilidades se representan por un *Modelo de Error*. Este Modelo de Error es más complejo que el anterior ya que permite definir operaciones de error que no son posible definir mediante un simple suavizado. En él se definen tres operaciones de edición: *sustitución* (incluyendo la sustitución de un símbolo por si mismo), *inserción* y *borrado*. Dados dos símbolos s_1, s_2 y el símbolo que representa al vacío ε, las sustituciones, inserciones y borrados son transducciones del tipo s_1/s_2, ε/s_2 y s_1/ε respectivamente. Además, mediante la combinación de estas operaciones básicas resulta sencillo definir otras operaciones más complejas (por ejemplo la transposición).

A cada una de estas operaciones se le puede asignar una probabilidad. La probabilidad de sustitución se deriva directamente de la matriz de confusión entre símbolos del proceso de entrada. Esta matriz consiste en una tabla que contiene las probabilidades de confusión de cada par de símbolos, y se estima a partir de un corpus representativo. Por ejemplo, si la entrada es un reconocedor de texto manuscrito, se modelan las probabilidades de confusión entre pares de caracteres. Si el proceso de entrada es un teclado, se puede representar las probabilidades de que un usuario pulse una tecla adyacente a la deseada en lugar de la correcta, etc. La matriz de confusión también es capaz de representar a los teclados en los que una misma tecla representa a varios caracteres, como ocurre en un teléfono móvil.

El modelo EM puede ser interpretado como un modelo *estático* de la incertidumbre del subsistema de entrada, complementando así a la estimación *dinámica* ofrecida por el modelo de hipótesis.

Las probabilidades de inserción y borrado de símbolos son dependientes de la tarea y pueden ser estimadas empíricamente. La figura 3.4 muestra ejemplos de tres transductores diferentes para el modelado del error en una tarea cuyo alfabeto lo forman los símbolos {a,b}. Estos transductores convierten una cadena x en una cadena y con probabilidad $p(x, y)$ o $p(y|x)$, según convenga, por medio de inserciones, borrados y sustituciones. Si un transductor de este tipo se compone por la derecha con un modelo HM ($HM \circ EM$), entonces cualquier hipótesis h producida por HM puede convertirse en $\hat{h}$, donde $\hat{h}$ es la versión ruidosa de h. Con $\Sigma = \Delta = \{a, b\}$. Este modelo transduce cualquier cadena en Σ^* a cualquier otra cadena en Δ^*.

La posibilidad de modelar el error mediante el uso de un transductor independiente facilita la construcción de éste al centrarse únicamente en tal modelización y no verse influido por otro tipo de interacciones. Así, resulta sencillo construir un modelo EM que permita operaciones de inserción, borrado y sustitución en cualquier lugar de la cadena o un autómata que permita la inserción de símbolos únicamente al principio de la cadena, véanse autómatas de la figura 3.4 superior izquierdo y derecho respectivamente. En el autómata inferior de la figura 3.4 se muestra otro ejemplo que permite los borrados sólo al principio o al final de una cadena y se limita el número de operaciones de error relativas a la transposición de dos símbolos adyacentes, véase el autómata inferior de la figura 3.4. Este último autómata podría ser de utilidad en el caso de tareas mecanografiadas donde existe bastante probabilidad de confusión de una tecla con otra bastante cercana.

3.2.4. El Modelo de Interacción con el Usuario (IM)

Cuando la intervención humana se hace necesaria, el proceso de decisión debe de incluir la opción de realimentación con objeto de incrementar la productividad durante dicho proceso de interacción. La naturaleza modular del método propuesto hace relativamente sencilla la incorporación de nuevos modelos que contengan información adicional. En este caso, el objetivo consiste en que el sistema pueda aprovechar la realimentación que el usuario ofrece al introducir nuevas salidas en tiempo real. La idea se basa en que conforme éste vaya introduciendo nuevos símbolos (habitualmente un prefijo de la salida deseada) el sistema vaya ofreciendo nuevas salidas compatibles con lo que ya se tenía junto a las restricciones que dicho usuario va imponiendo al sistema. Es de esperar que la introducción de la información interactiva, que el usuario ofrece en el modelo, redunde en una reducción del esfuerzo de éste, pues al añadir dicha información no es necesario el introducir completamente la cadena de salida. Además este método es fácilmente adaptable a otro tipo de interacción, como por ejemplo el habla, pues ésta puede ser fácilmente representada mediante una máquina de estados finitos.

En el caso más simple, en el que los caracteres que se introducen representan un prefijo de la cadena verdadera, un método sencillo para poder implementar el modelo de interacción

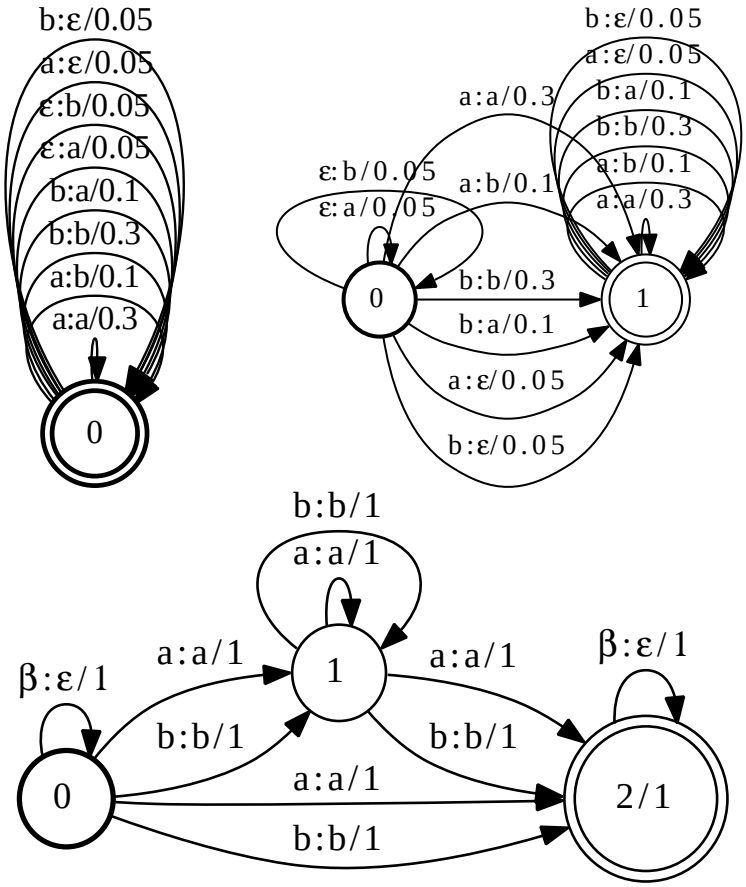

Figura 3.4: Ejemplos de tres transductores Modelo de Error (EM), con todas las posibles inserciones, borrados y sustituciones (superior izquierda), con inserciones permitidas únicamente al principio de la cadena (superior derecha) y con borrado de símbolo genérico, β, al principio y final de la cadena (inferior) y limitación de transposición de dos símbolos adyacentes.

consiste en lo siguiente: dado un prefijo $P = p_1, p_2, \ldots, p_n$, introducido por el usuario, se construye un transductor identidad con $n + 1$ estados al que se añade una transición entre cada par de estados s_i y s_{i+1}, asignando en cada una de estas transiciones el símbolo de entrada y salida p_i correspondiente del prefijo, a ese nivel de profundidad y con probabilidad 1. En el último estado s_{n+1} se añaden transiciones en forma de ciclo con todos los símbolos de entrada y salida $\sigma \in \Sigma$, donde Σ representa el conjunto de símbolos de la tarea. Este transductor *acepta/produce* el lenguaje de todas las cadenas posibles que se pueden formar con los símbolos de Σ que comiencen por el prefijo P. La figura 3.5 muestra el ejemplo de un transductor de este tipo que representa al prefijo $P = ab$.

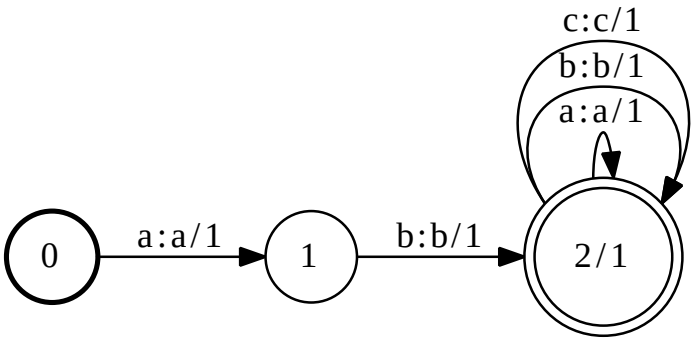

Figura 3.5: Ejemplo de un transductor identidad que representa al prefijo $P = ab$ y el alfabeto $\Sigma = a, b, c$.

El efecto de realizar una composición por la derecha de este transductor con el compuesto anteriormente, HM∘EM, (HM∘EM∘IM), consiste en el *filtrado* de las cadenas producidas por HM ∘ EM sin alteración de las probabilidades, de modo que sólo se dejan *pasar* aquellas que comienzan por el prefijo $P = ab$. En el apartado 3.2.5, aparecen más detalles acerca de esta composición. El tipo de interacción requerida por parte de un sistema es fácilmente modelable a través de este modelo IM. Así pues, si el tipo de interacción requerido por el sistema no es exactamente mediante prefijos, sino que es mediante sufijos, infijos o cualquier subcadena de la cadena deseada, se podría modificar dicho modelo IM con objeto de permitir otro tipo de modelos de interacción. Además de la posibilidad de poder implementar distintas estrategias de interacción, la ventaja del método propuesto es que permite combinar el resultado de la interacción con toda la información previa, lo que permite converger a la cadena correcta con un menor esfuerzo por parte del usuario.

Cuando la interacción con el subsistema no es lo suficientemente precisa, (como podría ocurrir en dispositivos móviles, donde el pequeño tamaño del teclado puede hacer que se produzcan muchos errores al teclear), se puede plantear una generalización de este modelo de interacción, que permita la tolerancia a errores al teclear. Para solucionar este problema, se debe asociar un modelo de error al modelo de interacción con el usuario (IM), de la misma forma que se ha hecho con el modelo HM. En este caso, habría un modelo EM asociado al modelo HM (EM_{hm}), representando los posibles errores que pueden ocurrir en la hipótesis y también existiría un modelo EM asociado al modelo IM (modelo de error asociado a la interacción con el usuario (EM_{IM})) que representaría los posibles errores debidos al proceso

de interacción. El modelo de error asociado al modelo de hipótesis (EM_{HM}) puede verse como un modelo estático de error cuyas transiciones están diseñadas especialmente para una tarea dada, mientras que el modelo de error EM_{IM} puede verse como un modelo de error relativo al tipo de errores que se producen durante la interacción.

Con frecuencia, los errores cometidos haciendo uso de un teclado por parte del usuario durante el proceso de interacción, consisten principalmente en la pulsación de teclas adyacentes a la deseada, y en menor medida, a la introducción de caracteres extra o bien a saltarse caracteres. Por lo tanto, el Modelo de Interacción con el Usuario (IM), debería de extenderse con transiciones que permitiesen aceptar prefijos *similares*, pero no necesariamente iguales al prefijo tecleado. La probabilidad de aceptación de un símbolo diferente al tecleado podría ser una función relacionada con la distancia física entre las teclas correspondientes (carácter pulsado y carácter aceptado) en el teclado, mientras que las probabilidades de las transiciones relacionadas con las inserciones y borrados podrían estimarse empíricamente. También podrían considerarse otras fuentes de errores durante la interacción, como pueden ser los errores de ortografía o ambigüedades. Por ejemplo, si el usuario teclea el prefijo $P = ab$ y se asume que a, b y c corresponden a teclas adyacentes en un teclado (con b entre a y c), se podría usar el modelo IM mostrado en la figura 3.6, que representa la interacción con el usuario compuesta con su modelo de error. Si el transductor es compuesto por la derecha con HM $\circ$ EM, entonces todas las cadenas producidas con los símbolos a, b, c serán aceptadas, pero las más probables serán aquellas que comiencen por ab. Este transductor es una extensión del mostrado en la figura 3.5 con transiciones extra: arcos etiquetados con $\varepsilon : s$ (con s en a, b, c), que permiten añadir símbolos extra al prefijo, arcos del tipo $s : \varepsilon$, que permite el borrado de símbolos desde el prefijo, y arcos del tipo $s : s$ que permite aceptar cualquier símbolo tecleado, o cualquier otro relacionado aunque con menor probabilidad, dependiendo dicha probabilidad de la función de distancia entre teclas considerada.

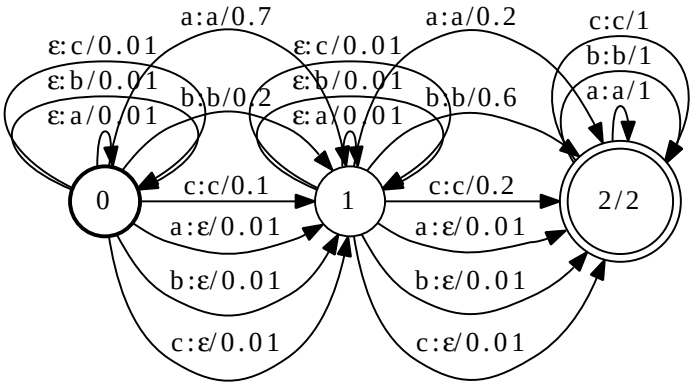

Figura 3.6: Ejemplo de un Modelo de Interacción con el Usuario que representa al prefijo $P = ab$, con transiciones para la recuperación de errores.

Una de las ventajas que la técnica de WFST presenta, es que los modelos propuestos

permiten diferentes formas de interacción, (tipografiando, escribiendo, hablando, etc.), exigiéndose únicamente que la entrada pueda representarse por medio de un transductor probabilístico, lo que permite una fácil integración de la multimodalidad.

En cuanto al coste computacional de introducción de dicho modelo IM, se puede decir que es similar al presentado en Llobet et al. [2010a] que crecía de manera sublineal con el incremento del tamaño del modelo de restricciones aplicando la composición *lazy*. El coste computacional del método propuesto no se ve excesivamente influenciado por el tamaño del modelo de restricciones, con lo que se puede decir que la cadena resultante puede estar disponible casi de manera inmediata tras cada interacción del usuario con el sistema.

3.2.5. Composición HM, EM, IM y CM

Con toda la información codificada en HM, EM, IM y CM, se puede encontrar la cadena más probable, según la gramática y la información que contiene cada uno de los modelos expuestos anteriormente. Esto se consigue componiendo todos los modelos en el orden adecuado y buscando, en el transductor resultado, el camino más probable. La cadena que finalmente se ofrecerá como resultado se obtiene mediante la concatenación de los diferentes símbolos de salida representados en cada una de las transiciones presentes en este camino más probable.

Tal y como se comentó al principio del apartado 3.2, dados dos transductores T_1 y T_2, la operación de composición, $T_1 \circ T_2$, selecciona la intersección entre el conjunto de cadenas producidas por T_1 y el conjunto de cadenas aceptadas por T_2. Por lo tanto, en el transductor resultado, tras dicha operación de composición, los símbolos de entrada provienen de los mismos símbolos de entrada que habían en el transductor T_1 mientras que los símbolos de salida provendrán de los símbolos de salida que había a la salida en el transductor T_2 en aquellas aristas cuya intersección a nivel de símbolo de salida de T_1 con símbolo de entrada de T_2 era distinta de vacío. Los caminos de este nuevo transductor serán aquellos cuyos símbolos de salida del transductor T_1, en un momento dado durante la operación de composición, coincidían con aristas del transductor T_2 cuyo símbolo de entrada en dicha arista era el mismo. Así pues, y a modo de resumen, se puede decir que el transductor resultado, transduce las cadenas desde la entrada en T_1 a la salida en T_2. Si T_2 es un transductor identidad (los símbolos de entrada y salida son los mismos), entonces el transductor actúa como un filtro dejando las cadenas sin cambio alguno. Tras el proceso de composición de dos autómatas la probabilidad resultante se calcula de acuerdo a los pesos definidos en cada transición. Si T_1 se modela conforme a un autómata conjunto y produce la cadena x con probabilidad $p(x)$ y T_2 transduce x a y con probabilidad $p(y|x)$, entonces $T_1 \circ T_2$ transduce x a y con probabilidad $p(x, y) = p(x)p(y|x)$ y el resultado será markoviano. En el caso en el que T_2 sea también un autómata conjunto y transforme x en y con probabilidad $p(y)$, entonces $T_1 \circ T_2$ transduce x a y con probabilidad $p(x)p(y)$, que coincide con $p(x, y)$ únicamente cuando x e y son independientes. Tal y como se expone en el apartado 2.3.2, para que la operación de composición de autómatas resulte en un autómata markoviano, entonces el primero de los autómatas debería de ser modelado conforme a un autómata conjunto y el resto como condicionales, tal y como se expone en el teorema de la probabilidad total ($P(ABC) = P(A) * P(B|A) * P(C|AB)$).

Por otra parte, desde el punto de vista del problema a solucionar aquí, y puesto que lo que finalmente interesa es el camino con un menor peso, el hecho de que el autómata no sea markoviano no va a ser un problema, puesto que aunque la suma de todas las probabilidades de todos sus posibles caminos no valga la unidad, el peso de los caminos, que es lo que finalmente nos interesa, si que nos indica cuales son los más interesantes desde el punto de vista del objetivo perseguido.

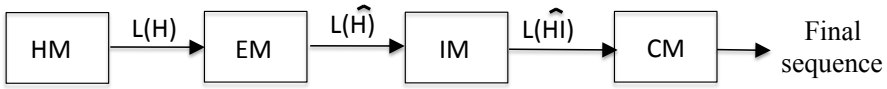

Figura 3.7: Composición de los diferentes transductores usados en el sistema propuesto.

Si se toma como referencia el orden de composición de los diferentes transductores que se muestra en la figura 3.7, lo primero que se calcula es la composición HM∘EM. Sea L(H) el conjunto de cadenas con pesos producidas por HM. Entonces, el transductor HM∘EM expande el conjunto de cadenas producidas por HM con nuevas cadenas, generalmente produciendo versiones perturbadas o ruidosas de la hipótesis original en L(H). A este nuevo conjunto de cadenas le vamos a llamar L($\hat{H}$).

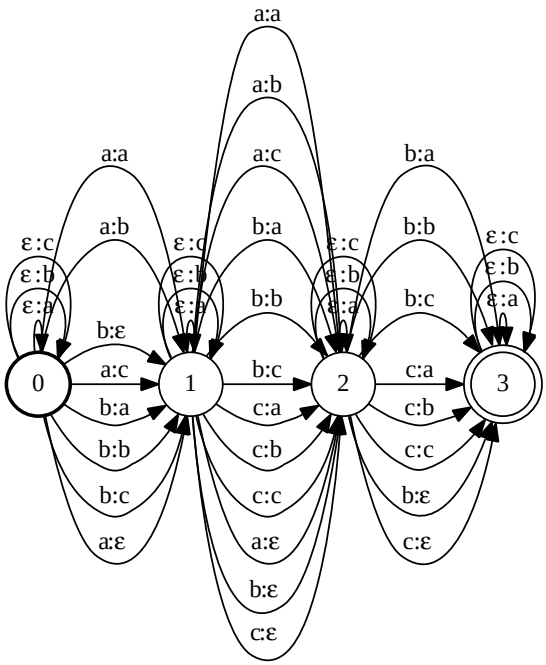

Figura 3.8: Composición del modelo HM mostrado en la figura 3.3 con un modelo EM que contiene todas las posibles sustituciones, inserciones y borrados (figura 3.4 izquierda) cuyo alfabeto Σ se ha ampliado con el símbolo c y el símbolo ε para las inserciones y borrados ($\Sigma = \{a, b, c, \varepsilon\}$).

La figura 3.8 muestra la composición de los transductores HM y EM mostrados en las figuras 3.3 y una versión ampliada ($\Sigma = \{a, b, c\}$) del autómata que aparece en la figura 3.4 superior izquierda. Este autómata transduce las cadenas producidas por HM a cualquier cadena en Σ^*, con una probabilidad más elevada conforme mayor sea la similitud entre la cadena original y la cadena transducida (en la figura 3.8 se han omitido las probabilidades por motivos de claridad).

Sea L($\hat{H}$) el conjunto de cadenas producidas por HM∘EM. Conforme el usuario interactúa con el sistema, se va creando el transductor identidad IM. Dicho transductor acepta y produce cadenas que contienen una subcadena *similar* (o igual en el caso de que IM no contenga transiciones de error) a la introducida por el usuario. El efecto de componer por la derecha IM con HM∘EM no es otro que seleccionar de L($\hat{H}$) aquellas cadenas compatibles con la gramática definida en IM y en el caso de que IM incluya transiciones de error, decrementar esta probabilidad conforme las diferencias con la entrada introducida por el usuario se incrementen. La figura 3.9 muestra la composición de los transductores HM∘EM de la figura 3.8 y el transductor IM de la figura 3.5. El resultado es un nuevo transductor que fija el prefijo introducido por el usuario. Alternativamente, se puede realizar la composición con un IM que incluya transiciones de error (figura 3.6). En este caso el resultado de la composición HM∘EM∘IM sería un autómata que genera cadenas con un prefijo *similar* al introducido por el usuario.

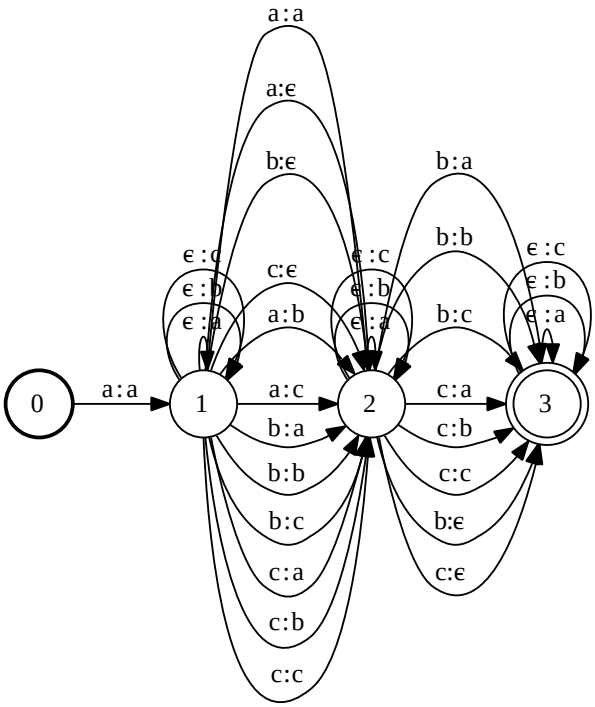

Figura 3.9: Transductor mostrado en la figura 3.8 con restricciones impuestas por el prefijo $P = a$ sin recuperación de errores.

Finalmente, sea L($\hat{H}$I) el conjunto de cadenas producidas por HM∘EM∘IM. El efecto de la composición por la derecha de CM con HM∘EM∘IM es la selección de aquellas cadenas de L($\hat{H}$I) aceptadas por el modelo de restricciones y el cálculo de los pesos de acuerdo a las probabilidades definidas en CM.

En resumen, el objetivo del método propuesto consiste en encontrar la transformación más probable de una cadena h, compuesta por la concatenación de los símbolos, $(h_1 \dots h_i \dots h_n)$, y producida por HM en una cadena s, compuesta por los símbolos, $(s_1 \dots s_i \dots s_n)$, y aceptada por el modelo CM por medio de las transducciones intermedias de los modelos EM y IM.[b] Formalmente, dada la probabilidad conjunta:

$$\hat{s} = \arg\max_{h,s} P(h, s) \tag{3.1}$$

donde

$$P(h, s) = P(HM, EM, IM, CM) \tag{3.2}$$

asumiendo independencia en los sucesos, HM,EM,IM,CM:

$$P(h, s) = P(HM)P(EM)P(IM)P(CM) \tag{3.3}$$

y como la influencia de cada uno de los modelos no es conocida:

$$P(h, s) = P(HM)^{\lambda_h} P(EM)^{\lambda_e} P(IM)^{\lambda_i} P(CM) \tag{3.4}$$

donde el peso asociado a cada λ, indica la importancia de la información procedente del modelo directamente asociado a ésta, respecto a la información del modelo de lenguaje CM que tiene un valor intrínseco de $\lambda_c = 1$.

Las probabilidades de cada uno de los sucesos, (HM,EM,IM,CM), presentes en esta ecuación se calcula a través de su transductor asociado. Conforme se ha podido ver anteriormente, el método buscará la cadena de salida, s, más probable a partir de la hipótesis inicial h, tomando en consideración el resto de evidencias intermedias. Para ello se buscará el camino más probable en el transductor HM∘EM∘IM∘CM resultante. Finalmente, la cadena $\hat{s}$ resultante, se construye a partir de la concatenación de los símbolos de salida.

Destacar que en este proceso, el único transductor que actualmente modifica los símbolos de la hipótesis original es EM, el resto de ellos son transductores identidad que dejan la cadena de entrada sin cambio alguno, siendo su único propósito la selección de un subconjunto de ellos con el fin de recalcular sus probabilidades. En el modelo propuesto no se ha realizado ningún tipo de análisis explícito, lo que es una clara diferencia con otro tipo de sistemas. En

[b] $h_i \in \{\Sigma \cup \varepsilon\}$ y $s_i \in \{\Sigma \cup \varepsilon\}$ lo que permitiría la aparición de inserciones y borrados y que la hipótesis h de entrada no tuviese porque tener el mismo tamaño que la hipótesis s de salida.

lugar de esto, lo que se propone aquí es la combinación de diferentes fuentes de información a través de modelos independientes en forma de autómatas WFST que se combinan mediante operación de composición de autómatas transductores.

Dado un transductor cuyos pesos representan a probabilidades, si todos los arcos de salida de un estado dado junto a la probabilidad de estado final tienen una probabilidad total 1, entonces la relación de entrada y salida, (e,s), que el transductor produce se denomina como relación conjunta y tiene una probabilidad $P(e, s)$. Por el contrario si todos los arcos de salida de un estado dado y un símbolo de entrada dado tienen una probabilidad total 1, entonces la relación se denomina condicional y tiene una probabilidad $P(s|e)$. Para que el resultado de la composición de autómatas WFST sea markoviano se requiere que una de las fuentes de información se modele como conjunta y el resto como condicional, conforme queda explicado en Eisner [2002]. Puesto que en nuestro caso, HM,IM y CM son autómatas transductores identidad y modelar cualquiera de estos autómatas como condicionales implicaría necesariamente la pérdida de parte de la información que ellos tienen, se opta por renunciar a conseguir la propiedad markoviana en el autómata final pues para el objetivo perseguido resulta preferible la fusión de toda la información existente en un momento dado, aún a costa de perder esta propiedad markoviana.

Tal y como se ha comentado anteriormente, como parte del proceso de búsqueda se requiere el cálculo del mejor camino. Un camino consiste en una secuencia, t, de n transiciones, $(t_1 \dots t_i \dots t_n)$, sobre el transductor compuesto y cada transición t_i lleva asociada una probabilidad que se calcula como el producto de las probabilidades de las transiciones correspondiente en HM, EM, IM, CM. Asumiéndose la independencia e igualdad de influencia entre todos los modelos, la probabilidad de una transición se define como:

$$\begin{aligned} P(s_i, h_i|t_i) &= P(HM, EM, IM, CM|t_i) \\ &= P(HM|t_i)P(EM|t_i)P(IM|t_i)P(CM|t_i) \end{aligned} \tag{3.5}$$

En la ecuación 3.6, la probabilidad de la cadena de salida se calcula como el producto de las probabilidades de las transiciones a lo largo del camino más probable en el transductor compuesto. Dado $x = (x_1 \dots x_i \dots x_n) \in L_1$ y $y = (x_1 \dots x_i \dots x_n) \in L_2$, la probabilidad de la transducción x,y es: $P(x, y) = \prod_{i=1}^{n} P(x_i, y_i|t_i)$, donde $t_1 \dots t_i \dots t_n$ es la secuencia de transiciones que transforman x en y.[c] Además, puesto que la influencia óptima de cada uno de los modelos es desconocida, se definen tres parámetros λ_h, λ_e y λ_i con objeto de obtener una combinación paramétrica loglineal de los modelos con diferentes pesos, conforme se vio en la ecuación 3.4:

$$\begin{aligned} P(s_i, h_i|t_i) &= P(HM, EM, IM, CM|t_i) \\ &= P(HM|t_i)^{\lambda_h} P(EM|t_i)^{\lambda_e} P(IM|t_i)^{\lambda_i} P(CM|t_i) \end{aligned} \tag{3.6}$$

[c] $x_i \in \{\Sigma \cup \varepsilon\}$ y $y_i \in \{\Sigma \cup \varepsilon\}$ lo que permitiría la aparición de inserciones y borrados y que el tamaño de x e y no tengan porque ser iguales.

Tal y como se puede apreciar en las ecuaciones 3.4 y 3.6, se asume un peso fijo de 1 para el modelo estático CM. Por lo tanto, su influencia se controlada por los valores absolutos del resto de parámetros. Los valores λ_h, λ_e y λ_i, junto con los costes de las operaciones de inserción y borrados (incluidas de manera intrínseca dentro del modelo EM) se estiman empíricamente utilizando un conjunto supervisado de entrenamiento y un algoritmo de optimización de parámetros, en nuestro caso dicho algoritmo ha sido el algoritmo *DownHill Simplex* Nelder and Mead [1965] que puede ser consultado con más detalle en el anexo A.1.

Para evitar problemas de desbordamiento, en lugar de trabajar con probabilidades se va a hacer uso del semianillo tropical WFST $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ donde $\mathbb{K}$ son logaritmos negativos de probabilidades, $\oplus$ es la operación *min*, $\otimes$ es $+$, $\bar{0}$ es $+\infty$ y $\bar{1}$ es 0. Por lo tanto, el camino más probable se encontrará buscando el camino de menor coste, que coincide con el camino más probable y para su búsqueda se hará uso del algoritmo Dijkstra [1959], donde puesto que el valor que tiene cada arista es siempre un valor positivo, no habiendo transiciones con costes negativos, se garantiza que el resultado final corresponderá a la cadena mas probable o de menor coste según se vea. El cálculo del coste en cada transición quedaría como se muestra en la ecuación 3.7

$$\begin{aligned}
\log(P(s_i, h_i|t_i)) &= \log(P(HM, EM, IM, CM|t_i)) \\
&= \log(P(HM|t_i)^{\lambda_h} P(EM|t_i)^{\lambda_e} P(IM|t_i)^{\lambda_i} P(CM|t_i)) \\
&= \log(P(HM|t_i)^{\lambda_h}) + \log(P(EM|t_i)^{\lambda_e}) + \\
&+ \log(P(IM|t_i)^{\lambda_i}) + \log(P(CM|t_i)) \\
&= \lambda_h \log(P(HM|t_i)) + \lambda_e \log(P(EM|t_i)) + \\
&+ \lambda_i \log(P(IM|t_i)) + \log(P(CM|t_i))
\end{aligned} \tag{3.7}$$

En el anexo A.6 se puede encontrar una interpretación más detallada de los parámetros que forman parte de un modelo loglineal como el que se ha propuesto en 3.7.

3.3. Un escenario práctico

Para ilustar estas ideas, se plantea un ejemplo sencillo basado en un escenario práctico. Imaginemos que tenemos el objetivo de procesar un campo concreto en un formulario que debe contener el nombre de un animal, y únicamente se permiten en dicho campo las entradas {*cat, cow, bat, goat*}. En estas condiciones nuestro modelo CM será el que se muestra en la figura 3.10. Por otra parte, si la salida OCR consiste en la cadena *aat*, el modelo HM que representa dicha salida será el que se muestra en la figura 3.11 (superior). Además, el hecho de que el clasificador OCR ofrezca, en algunos casos, varias hipótesis por símbolo, junto con sus probabilidades *a posteriori*, podría ser modelado: por ejemplo [a:1] para el primero de los símbolos, [a:0.6, o:0.4] para el segundo y [t:0.8, d:0.2] para el último, en este caso HM quedaría representado tal cual se muestra en la figura 3.11 (inferior). Finalmente, se asume

por simplicidad que nuestro alfabeto se restringe a los símbolos Σ ={a,b,c,g,o,t,w} y se conoce (de acuerdo a algunas muestras empíricas previas) que las probabilidades de confusión entre símbolos asociados con el clasificador OCR (matriz de confusión) y las probabilidades de inserción y borrado de símbolos son las que se muestran en la figura 3.12 (izquierda), con lo que el modelo EM consistirá en el transductor que se muestra en la figura 3.12 (derecha) (donde, por motivos de legibilidad, únicamente se muestran los arcos correspondientes a las confusiones mostradas en negrita en la tabla de la izquierda). Este modelo representa las operaciones de edición (sustitución, inserción y borrado) permitidas durante la transformación de la hipótesis inicial en una cadena válida. Aunque la matriz de confusión mostrada en el ejemplo contiene un número de ceros correspondientes a transformaciones no vistas, es usual suavizarla con objeto de permitir la sustitución de cualquier par de símbolos, pues se suele considerar que dichas transformaciones son posibles aunque no hayan sido observadas durante el proceso empírico de estimación de dicha matriz.

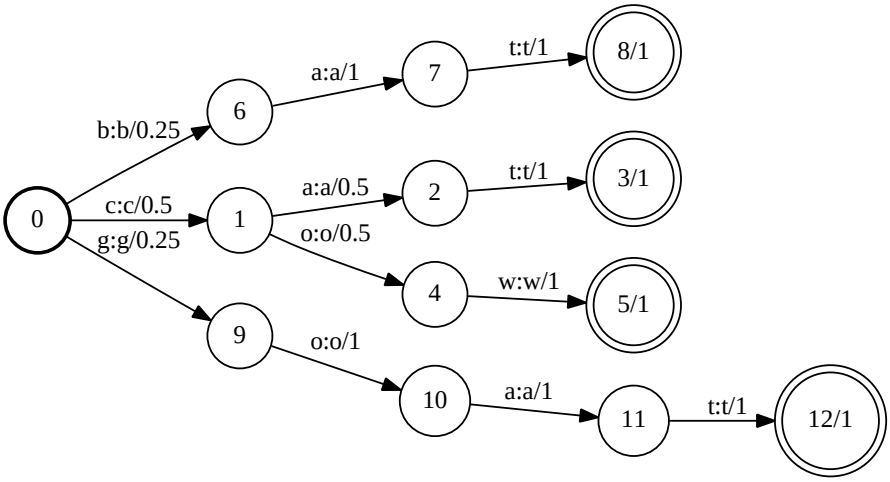

Figura 3.10: Modelo de restricciones que representa al diccionario {*cat, cow, bat, goat*}. Los estados con doble círculo son estados finales que incluyen la probabilidad de estado final en su interior.

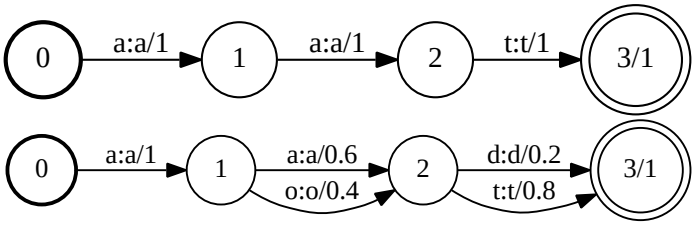

Figura 3.11: Modelo de Hipótesis representando la cadena de salida OCR *aat* sin probabilidades *a posteriori* (arriba) y con más de una posible clase por carácter junto con sus probabilidades *a posteriori* (debajo).

Si todos estos modelos se componen de la manera apropiada, el camino más probable del

	a	b	c	g	o	t	w	ε
a	**0.7**	0	**0.1**	0	**0.2**	0	0	**0.1**
b	0	0.7	0.1	0	0.2	0	0	0.1
c	0.1	0	0.7	0	0.2	0	0	0.1
g	0	0	0.1	0.7	0.2	0	0	0.1
o	0	0	0.3	0	0.7	0	0	0.1
t	0	0.2	0.1	0	0	0.7	0	0.1
w	0.1	0	0.1	0	0	0	0.8	0.1
ε	**0.1**	0.1	0.1	0.1	0.1	0.1	0.1	0.1

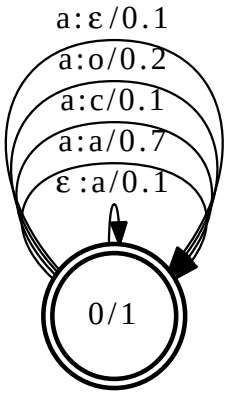

Figura 3.12: Ejemplo de matriz de confusión OCR junto con las probabilidades de inserción y borrado (símbolo ε), donde las filas representan a los símbolos de entrada y las columnas representan las salidas (izquierda) y su correspondiente Modelo de Error (derecha). Por motivos de legibilidad, en el modelo de la derecha tan solo se muestran las probabilidades destacadas en negrita en la matriz de la izquierda.

transductor resultante nos ofrecerá la transformación más probable de la hipótesis de partida del modelo HM en la cadena del modelo de restricciones CM a través de las operaciones de error definidas en el modelo EM, donde los símbolos de entrada/salida a lo largo del camino más probable representarán la secuencia de operaciones necesarias a la hora de transformar una cadena en otra, y el producto de las probabilidades en cada transición, la probabilidad de la transformación. En nuestro ejemplo, {*a/c, a/a, t/t*}, esto es, *aat* $\rightarrow$ *cat*, con probabilidad $p = 0{,}05 \times 0{,}21 \times 0{,}56 = 0{,}00588$. La tabla 3.1 nos muestra con detalle las diferentes operaciones realizadas.

Tabla 3.1: Transducción más probable para corregir la entrada *aat*

	HM produce	EM consume/produce	CM consume	Transformación
PASO 1	a:1	a/c:0.1	c:0.5	a/c:0.05
PASO 2	a:0.6	a/a:0.7	a:0.5	a/a:0.21
PASO 3	t:0.8	t/t:0.7	t:1	t/t:0.56

Tal y como se ha comentado anteriormente, si la cadena resultado no es correcta, puede ser eficientemente reestimada incorporando para ello la propia interacción del usuario en el modelo. Un método conveniente de interacción para el usuario consiste frecuentemente en la introducción del prefijo de la cadena esperada. Por ejemplo, imaginemos que la palabra correcta es *goat*, el usuario debería de introducir secuencialmente, de izquierda a derecha, los caracteres de esta palabra hasta que el sistema produjera la cadena de salida correcta. En este caso, si el usuario introduce el carácter *g*, el modelo IM que representa dicho prefijo sería el que se muestra en la figura 3.13. Este modelo tan solo acepta (y produce) cadenas

que comienzan con *g*, por lo tanto, si se incorpora entre los modelos EM y CM, actuará como un filtro que bloqueará cualquier cadena en la salida de EM que no comience con el prefijo representado. En este caso, la transformación más probable (actualmente la única posible) sería {ϵ/g, a/o, a/a, t/t}, esto es, *aat* $\rightarrow$ *goat*. La tabla 3.2 nos muestra las operaciones realizadas con un mayor detalle.

Tabla 3.2: Transducción más probable para corregir la entrada *aat* asumiendo el prefijo *g* como evidencia

	HM prod.	EM cons./prod.	IM cons./prod.	CM cons.	Transformación
PASO 1	-	ϵ/g:0.1	g/g:1	g:0.25	ϵ/g:0.025
PASO 2	a:1	a/o:0.2	o/o:1	o/o:1	o/o:0.2
PASO 3	a:0.6	a/a:0.7	a/a:1	a:1	a/a:0.42
PASO 4	t:0.8	t/t:0.7	t/t:1	t:1	t/t:0.56

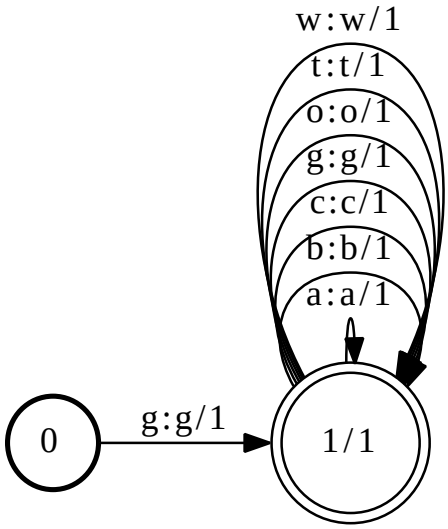

Figura 3.13: Modelo de Interacción que representa al prefijo {*g*} sobre el alfabeto Σ ={*a,b,c,g,o,t,w*}.

En resumen, el sistema propuesto tiene un doble propósito: transformar por un lado la evidencia observada (HM) en una cadena válida basándose en una aproximación por máxima verosimilitud y, en el supuesto de que la cadena propuesta no resulte ser correcta, obtener la cadena final con el mínimo esfuerzo por parte del usuario, añadiendo a las evidencias anteriores la propia evidencia resultado de la interacción con el usuario durante la corrección de la cadena.

Parte I

Corrección de cadenas

CAPÍTULO 4

POSTPROCESO DE OCR MEDIANTE EL USO DE WFST

Índice del capítulo

4.1. Objetivos

Una de las tareas que requieren más recursos en la industria de la digitalización de datos es el reconocimiento óptico de formularios manuscritos, donde se rellenan manualmente un gran número de campos preimpresos con distinta información en cada uno de ellos. La cantidad de trabajo que esto requiere es frecuentemente excesiva, y una pequeña mejora puede conllevar importantes ahorros. El postproceso aplicado a las cadenas resultantes del reconocimiento OCR, busca mejorar estas cadenas con objeto de reducir el esfuerzo que es necesario realizar para obtener su digitalización correcta. En el capítulo que ahora nos ocupa, vamos a realizar dicho postproceso mediante la aplicación de autómatas WFST y la operación de composición. El resultado final será un autómata que dadas unas hipótesis iniciales, procedentes del reconocimiento OCR, realizará y ponderará las diferentes operaciones de edición que permiten la transformación de dichas hipótesis, en cadenas admisibles por el modelo de restricciones previamente establecido para un campo dado. Tras esto, una búsqueda de la cadena más probable dentro de este autómata nos ofrecerá: 1) la cadena más parecida a la hipótesis de entrada, desde el punto de vista de las operaciones de edición, que pertenece al modelo de restricciones y 2) un valor relacionado con el coste de las operaciones de transformación que han tenido que realizarse para obtener dicha cadena final. Así pues, cuanto mayor sea dicho coste de transformación, mayor será el esfuerzo realizado en las operaciones de edición y por lo tanto mayor será la probabilidad de que la cadena finalmente corregida sea incorrecta. Es gracias a este coste de transformación que podemos establecer un umbral de aceptación/rechazo de cadenas, de tal manera que si una cadena tiene un coste de transformación por debajo de dicho umbral será considerada como válida automáticamente, también denominada positiva (**P**), mientras que si su coste está por encima será considerada inválida, también denominada negativa (**N**), y pasará a un proceso de validación posterior si el proceso así lo requiere. Esta clasificación de las cadenas, nos lleva a que una cadena corregida puede tener uno de entre cuatro estados, mutuamente excluyentes entre si, tal y como se detalla en la tabla A.4 del anexo A.5, y que son:

- **Verdadero positivo (VP)**: cadena correctamente corregida que ha sido clasificada como tal.
- **Falso positivo (FP)**: cadena incorrectamente corregida que ha sido clasificada como correcta.
- **Verdadero negativo (VN)**: cadena incorrectamente corregida que ha sido clasificada como tal.
- **Falso negativo (FN)**: cadena correctamente corregida que ha sido clasificada como incorrecta.

En este tipo de tareas, el principal requerimiento es una tasa de error (% FP) pequeña a nivel de campo, frecuentemente por debajo del 1 % o 0.5 %, para una tasa de rechazo dada. Los ahorros son directamente proporcionales a 1 menos la tasa de rechazo, anteriormente mencionada.

4.2. Introducción

Para obtener porcentajes significativos de aceptación con tasas de error muy bajas es necesario el uso de restricciones y conocimiento a priori sobre el contenido de los campos. Los modelos de lenguaje, usados con mayor frecuencia en texto libre, explotan el modelado de lenguaje léxico, sintáctico, semántico, pragmático y discursivo. En el caso de campos manuscritos las restricciones más frecuentes son léxicas (sólo se acepta un número limitado de posibles cadenas) o sintácticas (redundancia, códigos de control que son incluidos en la identificación, códigos de número de pasaporte o cuentas bancarias, etc.). Los modelos utilizados a nivel de campo, utilizan normalmente búsquedas en diccionarios, *n*-gramas, técnicas basadas en la distancia de edición, Modelos Ocultos de Markov y otros modelos de transición de categorías de palabras o caracteres. En el trabajo que aquí se presenta se muestran los resultados obtenidos mediante el uso del Análisis Corrector de Errores (ECP) Perez-Cortes et al. [2000] y los Transductores de Estados Finitos (WFST) Llobet et al. [2010a].

Cualquier método de reconocimiento óptico de texto escrito o impreso está sujeto a errores. Esto hace necesaria la aplicación de algún tipo de algoritmo de corrección sobre dicha salida con objeto de mejorar la tasa de reconocimiento. Los transductores de estados finitos (WFST) han sido utilizados de manera amplia en el reconocimiento del habla, la traducción automática y el reconocimiento de formas, entre otras disciplinas. Aquí se propone, aprovechar la flexibilidad de los WFST a la hora de modelar diferentes fuentes de información relativa a las probabilidades a posteriori del clasificador, la confusión entre símbolos y el modelo de restricciones para abordar dicho postproceso OCR. Los resultados obtenidos nos muestran que la combinación de las diversas fuentes de información mejoran el resultado final de postproceso, incrementando así la calidad y la productividad en un proceso de digitalización masiva de formularios. En el capítulo que aquí nos ocupa se van a mostrar los resultados obtenidos tras dicho postproceso de corrección OCR, tanto en campos aislados 4.3, como en campos con interdependencia 4.4.

4.3. Postproceso del reconocimiento OCR de campos aislados usando WFST

Lo habitual, en lo relativo al contenido que se puede introducir en los distintos campos que conforman un formulario, suele ser una lista de palabras extensa pero finita. Así pues, teniendo en cuenta que tan solo es admisible alguna de las palabras presentes en la lista y dada toda la información restante relativa al proceso de reconocimiento y corrección de la que se posee en un momento dado, existen métodos que permiten transformar una salida ruidosa de OCR en una cadena que pertenezca a un lenguaje, teniendo en cuenta además el resto de informaciones de que se disponga en ese mismo momento. El tipo de información más habitual del que se puede disponer en un sistema de digitalización OCR, consiste en: 1) las *probabilidades a posteriori del clasificador OCR*, 2) la *probabilidad de confusión entre símbolos* y 3) el *modelo de restricciones o de lenguaje*. La aproximación que aquí se propone conlleva la construcción y composición de varios WFST independientes que codifican cada

una de las distintas informaciones presentes y que han sido citadas anteriormente. Para ello se representa a cada una de ellas a través de distintos modelos individuales, extendiéndose así la idea del modelado de lenguaje OCR a través de la corrección estocástica de errores mediante análisis de la cadena comentada en Perez-Cortes et al. [2000].

En la tarea de postproceso de OCR se han identificado tres fuentes de información básicas: a) la salida OCR incluyendo todas las hipótesis para cada carácter junto con su probabilidad *a posteriori*, b) un modelo de los errores esperados, relativos a la confusión entre símbolos, inserciones y borrados, junto con sus probabilidades, y c) el lenguaje de las cadenas de la tarea a la que pertenece el campo en cuestión dentro del formulario. Cada una de estas fuentes de información puede ser representada mediante una máquina transductora de estados finitos estocástica a la que denominaremos respectivamente como Modelo de Hipótesis (HM), véase subsección 4.3.1, el Modelo de Error (EM), véase subsección 4.3.2, y el Modelo de restricciones o de Lenguaje (CM), véase subsección 4.3.3.

La entrada a corregir procede de un clasificador OCR, que en el caso que aquí nos ocupa está basado en un clasificador k-NN aproximado, construido a partir de imágenes de caracteres, las cuales han sido submuestreadas a un tamaño de 14x14 valores de gris. El número de dimensiones de los niveles de gris han sido reducidos, por medio de un análisis de componentes principales (PCA), a 45 dimensiones Pérez-Cortes et al. [2000]; Vidal et al. [2005]. Esto supone una reducción de la dimensionalidad de aproximadamente el 77 %. Se ha utilizado un valor de $k = 10$ y se han introducido todas las hipótesis ofrecidas por el clasificador que alimentan la etapa de postprocesado a través de la probabilidad a posteriori de la clase obtenida de acuerdo a la ecuación:

$$p(\omega_i \mid x) = \frac{\sum\limits_{j \in s_i} \frac{1}{d(x, y_j)}}{\sum\limits_{j=1}^{k} \frac{1}{d(x, y_j)}}$$

donde d es la *Distancia Euclídea* entre las características, x, de la muestra actual y la del prototipo respectivo, y_i. Y s_i, es el conjunto de subíndices de los prototipos de la clase w_i entre los k vecinos recuperados más cercanos, $y_1 \dots y_k$, Arlandis et al. [2002],

4.3.1. El Modelo de Hipótesis (HM)

Tal y como se ha comentado anteriormente en la sección 3.2.2, la salida de un clasificador OCR se puede ver, en su forma más general, como una secuencia de vectores n-dimensionales $\bar{v}_1 \dots \bar{v}_m$, donde n es el número de posibles hipótesis (clases) para cada carácter, m la longitud de la cadena de salida y $v_{i,j}$ la probabilidad a posteriori de la hipótesis j-ésima del carácter i-ésimo, cumpliéndose que $\forall i \sum_{j=1}^{n} v_{i,j} = 1{,}0$. Esta secuencia puede representarse mediante un WFSA (o un WFST identidad) que contendrá, $m + 1$ estados y n transiciones entre cada par de estados.

En el apartado 3.2.2 y concretamente en la figura 3.3 se muestra el ejemplo de un WFST, con alfabeto $[a, b, c]$, que representa la salida OCR con probabilidades a posteriori, ([0.8, 0.2, 0.0], [0.1, 0.7, 0.2], [0.0, 0.6, 0.4]), para cada posición de una cadena de longitud total igual a 3.

4.3.2. El Modelo de Error (EM)

Tal y como se comentó previamente en la sección 3.2.3, existen casos en los que ninguna de las secuencias de caracteres incluidas en las hipótesis OCR son compatibles con el modelo de restricciones (CM) final, o puede incluso darse el caso de que una variante similar sea más probable que cualquiera de las alternativas originales ofrecidas por el modelo HM. En un modelo clásico de n-gramas este efecto de caracteres no vistos es descontado por medio de un procedimiento de suavizado. En nuestro caso todas las posibles variaciones permitidas y sus probabilidades son representadas por medio de un Modelo de Error (EM).

En la sección 3.2.3 se definen tres operaciones típicas de edición: *sustituciones* (incluyendo la sustitución de un símbolo por si mismo, *inserciones* y *borrados*. A cada una de estas operaciones se les puede asignar una probabilidad. La probabilidad de las sustituciones se deriva a partir de la matriz de confusión del clasificador OCR. Dicha matriz puede verse como un modelo estático de la incertidumbre del clasificador OCR que complementa a la estimación dinámica que proviene de las propias probabilidades a posteriori ofrecidas por el clasificador. Las probabilidades de inserción y borrado son dependientes de la tarea y se han estimado, también de manera empírica, mediante un corpus de entrenamiento y un algoritmo de optimización como el mostrado en el anexo A.1.

La estimación de la matriz de confusión se realiza tras el propio proceso de creación del modelo OCR, haciendo uso de las muestras supervisadas utilizadas durante el entrenamiento de dicho modelo. Para ello, de cada muestra de entrenamiento se pedirá el número de vecinos con el que se desee trabajar más uno, $k + 1$. Evidentemente, y puesto que la muestra que queremos clasificar forma parte del propio modelo, el primer prototipo más cercano será la propia muestra a clasificar, por lo que este primer vecino es descartado. Una vez descartado el primero, se contabiliza de forma acumulada, y para cada uno de los prototipos de la muestra, el número de aciertos y fallos producidos en el resto de vecinos, asociando además las cuentas de los fallos con las clases de confusión. El resultado final de este proceso consiste, una vez se haya pasado por todos los prototipos de la muestra, en una matriz con las cuentas de aciertos y confusiones entre símbolos, a partir de las cuales se pueden hallar las probabilidades de confusión.

En el caso que aquí nos ocupa, en el autómata WFST resultado se van a permitir inserciones y borrados en cualquier parte de éste. Así pues, el WFST para el modelado del error (EM), que se aplicará finalmente, será similar al autómata superior izquierdo mostrado en la figura 3.4, pero añadiendo tantas combinaciones de símbolos, como resulten del producto cartesiano del alfabeto de entrada, cuyos símbolos provienen del modelo HM, con el símbolo ε, $\Sigma \cup \{\varepsilon\}$, para permitir así las inserciones, junto a los símbolos provenientes del alfabeto de salida, que proceden en este caso del modelo CM, con el símbolo ε, $\Delta \cup \{\varepsilon\}$, para permitir así los borrados.

4.3.3. El Modelo de Restricciones (CM)

El objetivo final de este modelo, es establecer las restricciones que han de tener las cadenas postprocesadas. Dichas restricciones van a configurarse en este trabajo como el conjunto de palabras, con o sin frecuencia absoluta, dependiendo de la tarea en cuestión, que forman parte del conjunto de cadenas aceptables por un campo en un formulario. Por otro lado, si se define una k en el lenguaje k-testable menor que la longitud de la palabra más larga, tendríamos conjuntos de n-gramas de símbolos, que permitirían lenguajes más abiertos. La manera de proceder del sistema de postproceso que aquí se expone, consiste en que ante una entrada ruidosa de OCR el sistema siempre ofrecerá como cadena de salida la cadena más próxima, es decir, aquella que tenga un coste mínimo de transformación y que sea aceptada por este modelo de restricciones. La forma en la que dicho modelo se expresa y se estima a través de un WFST ha quedado expuesta en la sección 3.2.1.

4.3.4. Postproceso de la cadena mediante la composición de los modelos CM, EM y HM

En el caso de corrección automática de cadenas no se ha considerado el incluir el modelo de interacción con el usuario (IM), expuesto en la sección 3.2.4, por su carácter automático y por lo tanto exento de cualquier interacción humana. Una vez se han modelado de forma aislada las tres fuentes de información expuestas anteriormente, éstas deberán ser combinadas de la manera adecuada para obtener el postproceso de corrección de cadenas planteado. La combinación de los diferentes modelos se realiza a través de la operación de composición entre transductores explicada formalmente en 2.2.

Sea L_1 el conjunto de cadenas que un modelo HM puede producir, en nuestro caso el conjunto de hipótesis del clasificador OCR, y L_2 el conjunto de cadenas aceptadas por un modelo CM dado, y que en el caso que nos ocupa sería el modelo de restricciones de un campo en un formulario. Nuestro objetivo es encontrar la cadena en L_2 más probable a partir de una cadena en L_1 por medio de la transducción definida en un modelo EM. Esto nos permite pasar de manera ponderada de L_1 a L_2. En el caso que nos ocupa la información en dicho modelo procede de una matriz de confusión entre símbolos y de unos pesos estimados para el caso de las inserciones y los borrados de símbolos. Este proceso de transducción anteriormente expuesto es equivalente a buscar el camino más probable, o de menor coste, en un nuevo transductor resultado de la aplicación de la operación de composición $(\circ)$ de estas tres fuentes de información en el orden que se expone a continuación, $HM \circ EM \circ CM$.

El transductor $T_1 = HM \circ EM$, transduce cualquier cadena de L_1, en un lenguaje más abierto que considera las posibles confusiones ponderadas entre símbolos aplicando a partir de la información de confusión modelada en EM. La figura 3.8 muestra el autómata resultado de la composición de los transductores HM mostrado en la figura 3.3 y el modelo de error, EM, mostrado en la figura 3.4 (superior izquierda) y al que se le ha añadido el símbolo c además del símbolo ε para las inserciones y borrados. Este nuevo autómata transduce las cadenas generadas por HM a cualquier cadena en Σ^*. Por lo tanto, el transductor $T_2 = T_1 \circ CM$, acepta únicamente las cadenas que pertenecen a L_2, y el resultado de la transducción

con el camino más probable es la cadena finalmente corregida. Si además se necesita obtener varias alternativas, se pueden obtener los n-mejores caminos de manera muy sencilla y de diversas formas. Una de ellas sería siguiendo los pasos del algoritmo 1:

Algorithm 1: Algoritmo para la extracción de los n-mejores caminos

Data:
N: Número de caminos requeridos.
*C*M: Modelo de restricciones.
*E*M: Modelo de error.
*H*M: Modelo de hipótesis.
Result: n-mejores caminos
k=0;
while $k \leq N$ **do**
  h = HM∘EM∘CM;
  $\hat{s}_k = \arg\max_{h,s_k} P(h, s_k)$;
  CM=CM-WFST($\hat{s}_k$) [a];
  k++;

[a]La complejidad de esta operación coincide con el de la operación de composición de autómatas, tal y como se comenta en la página web *http://www.openfst.org/twiki/bin/view/FST/DifferenceDoc* consultada el día 2/4/2015 estos costes son: **Coste Temporal:** $O(V_1V_2D_1(logD_2 + M_2))$ y el **Coste Espacial:** $O(V_1V_2D_1M_2)$, donde: V_i es el número de estados del autómata i, D_i es el número medio de transiciones que salen de un estado en el autómata i y M_i es la multiplicidad del autómata i, que es una medida del no determinismo, pues coincide con el número medio de veces que una etiqueta se repite en un estado.

4.3.5. Definición de coste y optimización de parámetros

Se define una cadena s como un conjunto de símbolos concatenados $s = \{s_1 \cdots s_i \cdots s_n\}$. Las cadenas pueden ser tanto de entrada como de salida. En el ejemplo que nos ocupa, considerando ϵ también como símbolo, cada símbolo de entrada va a tener asociado un símbolo en la cadena de salida, de esta manera cada transición en el autómata transductor coincide con uno de estos pares de símbolos de entrada/salida. Así pues, dadas dos cadenas: $x = \{x_1 \cdots x_i \cdots x_n\} \in L_1$ y $y = \{y_1 \cdots y_i \cdots y_n\} \in L_2$, la probabilidad del par (x, y), se calcula como el producto de las probabilidades de transición a lo largo del camino que genera la cadena de salida a partir de la cadena de entrada en el transductor compuesto, de tal manera que dado $x \in L_1$ y $y \in L_2$, la probabilidad de la transducción de x en y es $P(x, y) = \prod_{i=1}^{n} P(x_i, y_i|t_i)$, donde $t_1 \ldots t_n$ es la secuencia de transiciones que transduce la cadena de entrada, x, en la cadena de salida, y. Puesto que las fuentes de información en esta parte son: HM, EM y CM, la probabilidad de una transducción en una transición t_i de un símbolo x_i en un símbolo y_i, en el autómata transductor resultado de la composición, se define como la probabilidad conjunta de los tres modelos, $P\,(HM, EM, CM|t_i)$.

$$P(x_i, y_i|t_i) = P(HM, EM, CM|t_i) \tag{4.1}$$

que asumiendo independencia entre cada una de las fuentes de información:

$$P(x_i, y_i|t_i) = P(HM|t_i)P(EM|t_i)P(CM|t_i) \tag{4.2}$$

Puesto que la influencia óptima de cada uno de los modelos es desconocida, se definen además dos parámetros λ_e y λ_h, que dotarán de elasticidad al modelo y cuyos valores serán indicativos, conforme se ve en la sección A.6 del anexo final, de la importancia de la información del submodelo asociado al parámetro λ correspondiente, sobre el modelo general.

$$P(x_i, y_i|t_i) = P(HM, EM, CM|t_i) = P(HM|t_i)^{\lambda_h} P(EM|t_i)^{\lambda_e} P(CM|t_i) \tag{4.3}$$

Tal y como se aprecia en la ecuación 4.3, se asume un peso fijo de 1 para el modelo CM. Por lo tanto, su influencia es controlada por los valores absolutos del resto de parámetros, que dan cuenta de los pesos relativos del resto de modelos respecto del modelo CM. La estimación de los parámetros λ_e y λ_h, junto con las probabilidades de las operaciones de inserción y borrado, que están incluidos en el propio modelo EM y que fueron anteriormente mencionados en la subsección 4.3.2, son estimados empíricamente a través de un conjunto supervisado de entrenamiento. A partir de aquí se obtiene una combinación paramétrica *loglineal* de los modelos con diferentes pesos, pudiéndose expresar el cálculo del coste como:

$$\log P(x_i, y_i|t_i) = \lambda_h \log P(HM|t_i) + \lambda_e \log P(EM|t_i) + \log P(CM|t_i) \tag{4.4}$$

Como el logaritmo de valores positivos por debajo de 1 es un valor negativo, y los valores de probabilidad están acotados entre 0 y 1, se define la función de coste de corrección en una transición t_i, $c(x_i, y_i|t_i)$, como:

$$\begin{aligned} c(x_i, y_i|t_i) &= -\log P(x_i, y_i|t_i) \\ &= -\lambda_h \log P(HM|t_i) - \lambda_e \log P(EM|t_i) - \log P(CM|t_i) \end{aligned} \tag{4.5}$$

Ahora, tal y como se muestra en la ecuación 4.5, todos los costes serán positivos y además conforme su valor absoluto sea mayor significará un mayor coste en las operaciones de transformación. Ello estará directamente relacionado con el uso de operaciones de transformación que son menos probables y que por lo tanto incrementan dicho coste. A partir de aquí el coste total de corrección de una cadena x en una cadena y, cuya transformación ha requerido el paso por n transiciones, se calcula como:

$$\begin{aligned} c(x, y) &= \sum_{i=1}^{n} -\log P(x_i, y_i|t_i) \\ &= \sum_{i=1}^{n} -\lambda_h \log P(HM|t_i) - \lambda_e \log P(EM|t_i) - \log P(CM|t_i) \end{aligned} \tag{4.6}$$

El cálculo del mejor camino que existe en el autómata compuesto, desde un punto de vista probabilístico, es el elemento clave del proceso. Para evitar problemas de desbordamiento, en lugar de trabajar con probabilidades, vamos a usar en los WFST el semianillo tropical $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$, tal y como se expone en el capítulo 2, donde $\mathbb{K}$ son logaritmos de probabilidades negativas, conforme se muestra en la ecuación 4.6, $\oplus$ es la operación *min*, $\otimes$ es $+$, $\bar{0}$ es $+\infty$ y $\bar{1}$ es 0. Por lo tanto, el camino más probable se encontrará al buscar el camino con mínimo coste, lo que coincidirá directamente con la transformación más probable de la cadena de entrada en la cadena de salida.

En la industria de la digitalización masiva de formularios es muy importante la obtención de valores de confianza consistentes, entendiendo en nuestro caso dicha confianza como la probabilidad asociada al camino más probable en el transductor compuesto. Debido al semianillo utilizado en nuestro caso, este camino de máxima probabilidad coincidirá con el camino con menor coste de transformación de la cadena de entrada en la cadena de salida. Una vez obtenido este mejor camino, y para evitar el efecto que la longitud de la cadena tiene al producir, de manera general, costes más elevados en cadenas más largas, se utilizará como medida de confianza el coste medio de corrección, calculado a través de los costes obtenidos en cada una de las transiciones por las que pasamos para obtener este mejor camino. Este concepto de confianza en la corrección, permite al usuario elegir un umbral junto a una estrategia apropiada de rechazo, conforme se verá en el capítulo 5, que acepte el mayor número de cadenas correctas y rechace el mayor número de las incorrectas.

Los parámetros λ_i anteriormente mencionados, que ofrecen elasticidad al modelo, se optimizan siguiendo el criterio de una función objetivo que busca maximizar la tasa de reconocimiento. Dicha función objetivo se ha definido como el porcentaje, con respecto al conjunto total de test, de las cadenas que fueron aceptadas y exitosamente corregidas (VP) para una tasa de error dada, entendiendo dicha tasa de error como el porcentaje, definido también sobre el conjunto total de test, de las cadenas que fueron aceptadas como correctas pero realmente habían generado correcciones incorrectas (FP). Con esta estrategia, únicamente las cadenas rechazadas (N=VN+FN) han de ser revisadas por operadores humanos, lo que significa que para una tasa de error comercialmente aceptable, el ahorro económico llevado a cabo por el sistema es aproximadamente equivalente al número de cadenas automáticamente aceptadas por dicho sistema (P=VP+FP) y asumiendo que los FP representan a una tasa de error asumible. En el capítulo 5 se presenta un estudio sobre como asociar un error asumible por el usuario (% FP) con un coste de corrección en un modelo de lenguaje determinado. En el capítulo 6 se muestra un ejemplo de uso de los autómatas WFST con el objetivo de incrementar la productividad en la validación humana de las cadenas rechazadas automáticamente por el sistema (N) y que han de pasar obligatoriamente por dicho proceso de validación. Para la optimización de los parámetros del modelo anteriormente expuestos, bajo el criterio de la función objetivo comentada, se ha hecho uso del algoritmo de optimización *downhill simplex* que se explica con mayor detalle en el anexo A.1.

4.3.6. Composición *lazy*

La composición de WFST, puede incrementar el coste computacional cuando los transductores son grandes, pues tal y como se comenta en Mohri [2004] en el peor caso en el que todas las transiciones de T_1 que abandonan un estado q_1 se acoplen con las que abandonan T_2 en un estado $q_1^{'}$ entonces el coste computacional tendrá un orden cuadrático con el tamaño de los autómatas ($\approx O(|T_1||T_2|)$. Para el caso de un CM de 64000 estados y 140000 transiciones (como el utilizado en nuestros experimentos), un modelo EM con todas las posibles inserciones, borrados y sustituciones y un tamaño medio de HM con 8 estados y 5 transiciones (hipótesis) por estado, el transductor compuesto resultante puede tener por encima de 450000 estados y más de dos millones de transiciones, lo que redunda en un excesivo coste espacial y computacional a la hora de realizar dicha operación. Para evitar este problema, se ha usado la composición *lazy*. Las operaciones *lazy* retrasan el cálculo del resultado hasta que es requerido por otra operación. Esto es útil cuando un gran transductor intermedio debe ser construido pero sólo se va a necesitar visitar una parte muy pequeña de éste Mohri et al. [2000]. En la aproximación implementada la operación de composición se retarda hasta que se requiera la búsqueda del mejor camino en el transductor. Esto redunda en costes espaciales y temporales asumibles, pues dicha operación se aplica tan sólo sobre las partes del autómata que son requeridas en algún momento durante dicho proceso de búsqueda.

En principio, no es necesario completar la composición de transductores para calcular el mejor camino. Para buscar el mejor camino, se usa un algoritmo de primero el mejor (1-best) que explora el autómata expandiendo el camino de coste acumulado más bajo en cada estado Dijkstra [1959]. Ello permite encontrar el camino de menor coste sin necesidad de explorar el autómata entero.

4.3.7. Experimentos y resultados

En esta sección se detalla la metodología experimental y los resultados de la evaluación correspondiente de una realización particular de la corrección del sistema simbólico para ilustrar su funcionamiento y utilidad. Los siguientes experimentos comparan el sistema, a partir de una tarea OCR determinada, trabajando tanto con como sin múltiples hipótesis y probabilidades *a posteriori* en el modelo de hipótesis (HM). Para el experimento se ha utilizado una muestra de 14000 apellidos manuscritos a través de formularios escaneados en una tarea industrial real. El modelo de restricciones (CM) está formado por un modelo de lenguaje de referencia de 4.5 millones de apellidos (99157 de los cuales eran únicos). Se ha usado una $k = 25$ igual al tamaño del apellido más largo en el modelo de lenguaje, de tal manera que tan solo los apellidos conocidos en la lista son aceptados como salidas correctas. Se ha utilizado la librería OpenFST para los experimentos Allauzen et al. [2007]; Mohri et al. [2000].

El corpus se ha dividido en dos partes: un conjunto de entrenamiento (15 %) y otro de test (85 %). El conjunto de entrenamiento se usa para entrenar los parámetros del modelo de error (probabilidades de inserciones y borrados) y también los pesos asociados a la composición de los diferentes WFST que entran en juego (λ_h y λ_e). Se han realizado optimizaciones independientes para cada aproximación mediante el algoritmo de optimización mostrado en la

sección A.1 del anexo. La influencia de cada modelo individual puede variar dependiendo de la aproximación seleccionada, pues no es lo mismo utilizar múltiples hipótesis y probabilidades *a posteriori* para cada una de las hipótesis (WFST-PP) que utilizar sólo la entrada más probable (WFST).

En las tareas de proceso de formularios en la industria de digitalización de datos, es muy importante obtener valores consistentes de medidas de confianza que indiquen la fiabilidad de la cadena propuesta (en nuestro caso la probabilidad del camino más corto en el transductor combinado) permitiendo así al usuario definir un umbral y una adecuada estrategia de rechazo. Por ello, se han optimizado los parámetros utilizando una función criterio que maximiza la tasa de reconocimiento conforme se comenta en 4.3.5. Con esta estrategia tan solo las cadenas rechazadas por el sistema deberán ser revisadas por operadores humanos, lo que significa que para una tasa de error comercialmente aceptable (% FP), los ahorros económicos llevados a cabo por el sistema son casi equivalentes al número de cadenas automáticamente aceptadas como buenas por dicho sistema.

	λ_h	λ_e	p_i	p_d
WFST-PP	2.38	1.17	0.005	0.004
WFST	1.04	1.05	0.007	0.008

Tabla 4.1: Parámetros óptimos encontrados con y sin probabilidades *a posteriori*.

En la tabla 4.1 se muestran los mejores parámetros encontrados para WFST y WFST-PP. Es de destacar que el punto de trabajo óptimo para WFST se alcanza cuando todos los modelos tienen pesos similares (el modelo de lenguaje (CM) es un modelo de referencia que siempre tiene 1), mientras que la aproximación con WFST-PP alcanza su mejor rendimiento cuando el modelo de hipótesis tiene un peso mayor que el resto de modelos, debido a que en dicho modelo se incluyen las probabilidades a posteriori (PP) como información relevante, esto hace crecer la importancia de dicho modelo frente al resto de modelos (véase interpretación de los parámetros λ_i en la sección A.6 del anexo). Se observa también que las probabilidades de inserción y borrado son menores en la aproximación WFST-PP, puesto que bajo esta aproximación existe un mayor número de cadenas que pueden ser corregidas con un coste más pequeño simplemente eligiendo uno de los símbolos propuestos por el modelo de hipótesis (HM) en lugar de operaciones de inserción y borrado.

La figura 4.1 muestra las tasas de reconocimiento ($\frac{VP}{P+N}$) y de error ($\frac{FP}{P+N}$) del método propuesto usando: *a)* múltiples hipótesis y probabilidades *a posteriori* en el modelo de hipótesis (HM), WFST-PP, *b)* la misma aproximación pero haciendo únicamente uso de la cadena de entrada con una única hipótesis por carácter, WFST, y *c)* la cadena de salida del reconocedor OCR sin corregir. Los distintos puntos de la gráfica se obtienen variando el umbral de rechazo.

En la figura 4.2 se aprecia la mejora producida en la tasa de reconocimiento de WFST-PP frente a WFST, en función de la tasa de error. En el caso que aquí presentamos los costes de transformación presentes en las aristas son siempre 0 o positivos, y los tiempos de búsqueda

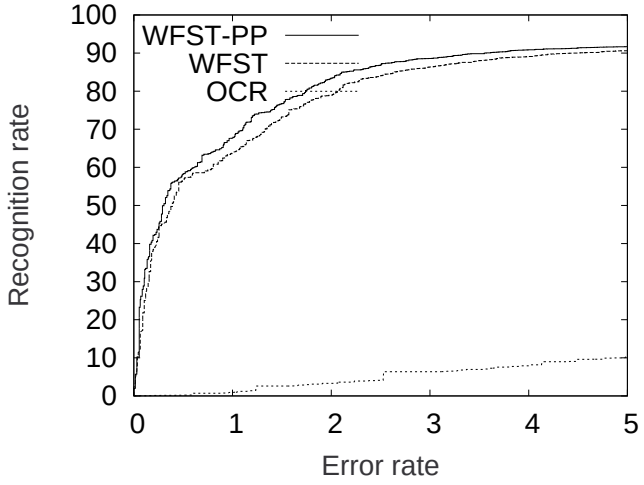

Figura 4.1: Tasas de reconocimiento frente al error

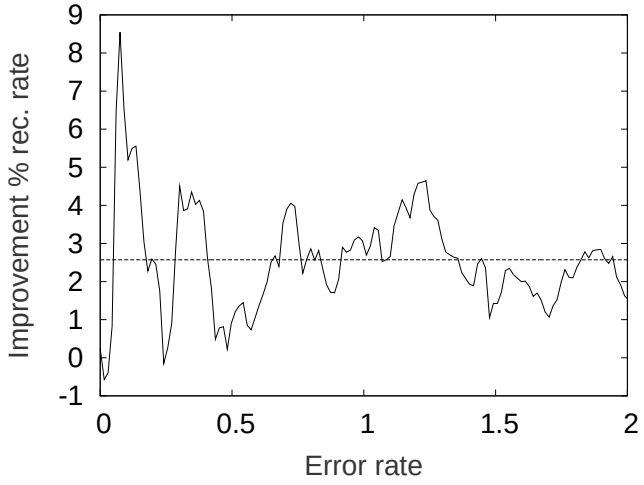

Figura 4.2: Mejora de la tasa reconocimiento del sistema WFST-PP frente a WFST en función del error

del mejor camino, en los autómatas compuestos que subyacen para este tipo de tareas, resulta ser del orden de milisegundos, en un Intel Xeon 2.5 GHz con 2GB de memoria, sistema operativo Linux OS y gcc4.4.

También se han realizados una serie de experimentos conducidos a comparar la aproximación propuesta, con el método previo Perez-Cortes et al. [2000] de análisis estocástico corrector de errores. Este método hace uso de un autómata estocástico de estados finitos, en lugar de WFST, a la hora de establecer el modelo de restricciones. Su objetivo consiste en

transformar una cadena de entrada, resultado de un clasificador, junto a su matriz de confusión, en la cadena más próxima del modelo de restricciones, haciendo uso para ello de las operaciones de edición anteriormente mencionadas. La principal diferencia de este método con el que presentamos en el marco de esta tesis reside en la utilización de autómatas WFST frente a los autómatas de estados finitos. El uso de WFST ofrece un marco común bien estudiado que simplifica enormemente la introducción de nuevas fuentes de información, pues cada una de ellas puede ser modelada de manera independiente, este modelado facilita enormemente la introducción de nuevas evidencias. Tras la creación de los WFST que modelan las diferentes fuentes de información éstos se componen de la manera apropiada, mediante la operación de composición de autómatas, produciendo con ello un único autómata completamente acoplado que lo integra todo. Es sobre este autómata sobre el que se ha de realizar la búsqueda del mejor camino que se corresponde con la cadena más próxima del modelo de restricciones dadas las evidencias de entrada y las confusiones entre símbolos.

Para la realización de los experimentos se parte del corpus previo dividido en entrenamiento (15 %) y test (85 %). Al igual que antes el corpus de entrenamiento se ha usado para estimar los parámetros del modelo de error (probabilidades de inserción y borrado) y los parámetros del modelo de hipótesis (λ_h) y el modelo de error (λ_e) en la composición de autómatas WFST.

Tasa de error(%)	Tasa de reconocimiento (%)			
	WFST-PP	WFST	ECP	OCR
0.1	30.6	25.1	24.7	0.1
0.25	45.5	40.5	41.2	0.6
0.5	61.5	54.6	53.9	0.7

Tabla 4.2: Resultados experimentales

La tabla 4.2 muestra la tasa de reconocimiento para tres tasas de error asumidas y cuatro aproximaciones diferentes: *a)* El método propuesto haciendo uso de múltiples hipótesis y probabilidades *a posteriori* en el modelo de hipótesis HM (WFST-PP), *b)* la misma aproximación usando únicamente una única hipótesis por carácter (WFST), *c)* análisis estocástico corrector de errores Perez-Cortes et al. [2000] (ECP) y *d)* utilizar la salida OCR sin corregir (OCR).

La figura 4.3 muestra el 10 % inicial de la curva ROC para estos experimientos.

4.3.8. Conclusiones

En esta sección se ha propuesto un método de postproceso de OCR que hace uso de autómatas transductores WFST a la hora de codificar diferentes informaciones, como pueden ser: el conjunto de hipótesis ofrecidas por el clasificador, el modelo de error de confusión entre caracteres y un modelo de restricciones, que se ha creado previamente, a partir de un algoritmo de inferencia de un lenguaje *k*-testable. Para la integración de toda la información aportada por los distintos modelos de manera independiente, se ha hecho uso de la operación

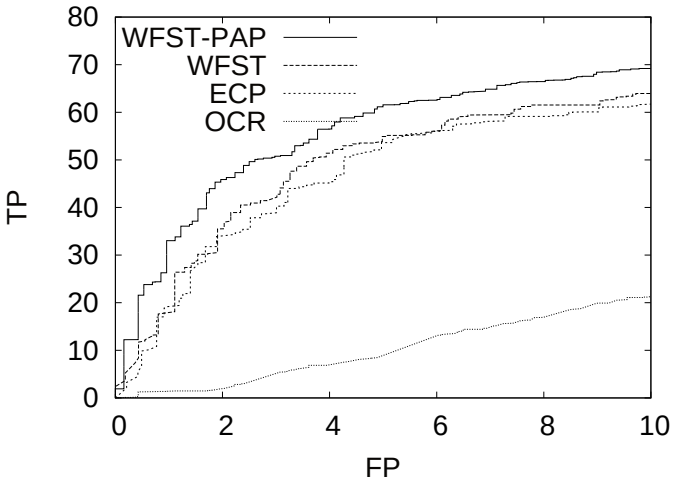

Figura 4.3: Curva ROC comparativa de diferentes aproximaciones, donde $TP = \frac{VP}{VP+FN}$ y $FP = 1 - \frac{VN}{VN+FP}$.

de composición de autómatas. La solución final al problema planteado pasa por la búsqueda del camino de mínimo coste en el transductor final compuesto. Este transductor es el resultado de la composición de cada uno de los modelos independientes. Este camino más probable nos ofrece la cadena más probable compatible con dicho modelo de restricciones, el modelo de hipótesis y el modelo de error planteados.

Puesto que la aplicación de la operación de composición es costosa, tanto espacial como temporalmente, se ha planteado una composición *lazy*. El algoritmo de Dijkstra [1959] bajo las condiciones de no negatividad de los pesos en los caminos de un autómata, y el tipo de autómatas deterministas y mínimos que finalmente se obtienen como resultado, garantiza una búsqueda óptima del camino de mínimo coste sin la necesidad de realizar la composición de autómatas de manera previa.

Finalmente y de acuerdo a los test llevados a cabo con datos manuscritos, se han obtenido mejoras significativas y de manera eficiente, sobre aproximaciones previas con otro tipo de metodologías. Por otra parte, se ha observado un incremento en la mejora de los resultados, corrigiéndose correctamente por encima de un 90 % de los apellidos tras aplicar el método de postproceso. Además se observa que la introducción de las probabilidades a posteriori en el modelo, ha producido un incremento significativo de las tasas de reconocimiento respecto a su no inclusión, véase las figuras anteriormente expuestas. Tras la optimización automática de los parámetros se ha observado un incremento en la importancia del modelo de hipótesis al incluir en éste las probabilidades a posteriori respecto a no incluirlas, pues en el caso de considerarlas el modelo contiene una información relevante que le hace ganar peso respecto del resto de modelos y eso se observa a través del valor del parámetro λ_h asignado de manera automática por el algoritmo de optimización en relación a los valores asignados al resto de parámetros (λ_e y $\lambda_l = 1$) asociados con los otros modelos, véase parámetros de la tabla 4.1.

También se observa un incremento significativo en la tasa de reconocimiento para tasas de error bajas, al hacer uso de las probabilidades a posteriori respecto a no utilizarlas, además de observar un comportamiento algo mejor al comparar el sistema propuesto sin probabilidades a posteriori con el sistema ECP que tampoco las consideraba como fuente de información, véase tabla 4.2.

Finalmente, hacer uso de un modelo de error, de hipótesis OCR y de restricciones que puedan ser modelados de forma independiente, junto a una metodología como la de WFST que permita integrarlos a través del uso de la operación de composición de autómatas, ofrece en la práctica importantes ventajas sobre otros paradigmas más acoplados pues permite la modificación e inclusión de los modelos sin afectar por ello a otras partes del sistema.

4.4. Postproceso del reconocimiento OCR haciendo uso de la interdependencia entre campos mediante WFST

Existen muchas ocasiones en las que aparece un grado de interdependencia entre los campos de un formulario. Ejemplos típicos serían: regiones, estados, provincias, ciudades, calles, códigos postales, etc. Estas interdependencias establecen nuevas restricciones que pueden ser utilizadas para mejorar la corrección. Para aprovecharla se definen los modelos de lenguaje combinados. En este apartado se van a testear dichos modelos combinados con los modelos WFST y se van a reportar las mejoras que se producen con la combinación de campos en comparación a su tratamiento como campos aislados.

4.4.1. Combinación de campos

Dada la flexibilidad del modelo WFST, existen diversas maneras de combinar los modelos de lenguaje de varios campos que por su naturaleza son interdependientes. Aquí se ha utilizado un procedimiento simple, basado en la concatenación de todas las posibles combinaciones de los campos permitidas haciendo uso de un símbolo de enlace especial que nos sirve para la separación de los diferentes campos.

De manera detallada, dados dos campos F_1 y F_2, donde se permiten cadenas del modelo de lenguaje (CM) L_1 y $L2$ respectivamente, se generan todas las posibles combinaciones válidas de cadenas en L_1 con cadenas en L_2, posicionando entre cada par de subcadenas, que forman una cadena combinada válida, un símbolo especial $+$. Esto genera el modelo de lenguaje $L_1 + L_2$. Tras esto, en la fase de ejecución, dados dos modelos de hipótesis hm_1 y hm_2 obtenidos a partir de las salidas OCR de los campos F_1 y F_2, se genera el modelo combinado $hm_1 + hm_2$ y se combina con el modelo de confusión entre símbolos (EM) y el modelo $L_1 + L_2$, con objeto de convertir las cadenas concatenadas del modelo de hipótesis en las cadenas concatenadas más próximas del modelo de lenguaje a través de las transducciones entre símbolos ofrecidas por el modelo de confusión.

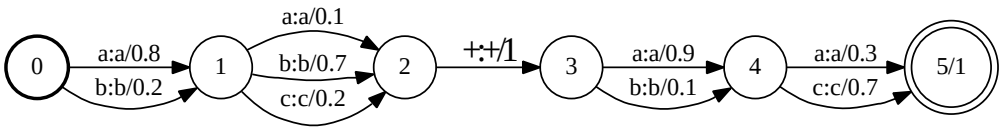

Figura 4.4: Ejemplo de la combinación de dos modelos de hipótesis.

En la figura 4.4 se muestra el ejemplo resultante de la combinación de dos modelos de hipótesis:

$$hm_1 = [a : 0{,}8, b : 0{,}2, c : 0{,}0]\,, [a : 0{,}1, b : 0{,}7, c : 0{,}2]$$
$$hm_2 = [a : 0{,}9, b : 0{,}1, c : 0{,}0]\,, [a : 0{,}3, b : 0{,}0, c : 0{,}7]$$

En el modelo de error, se impone la restricción de que el símbolo especial $+$ puede sustituirse únicamente por él mismo y no puede ser ni borrado ni insertado. Este símbolo especial

actúa como una barrera natural que evita que los símbolos procedentes de un campo migren al otro campo en cualquier punto del proceso de análisis. La cadena de salida corregida siempre tendrá un símbolo $+$ que será usado para dividir dicha cadena y asignar cada corrección a cada uno de los campos F_1 y F_2 a la que pertenecen.

4.4.2. Experimentos y resultados

Se han realizado una serie de experimentos para comparar los resultados de los modelos de campos combinados con modelos de campos individuales. El tipo de campos elegidos tienen diferentes grados de interdependencia. Todos los campos tienen modelos de lenguaje creados a partir de una lista de cadenas válidas. En ellos se ha usado una k, en un lenguaje k-Testable igual a la cadena más larga del corpus, lo que significa que el lenguaje tan sólo aceptará como válidas las cadenas presentes en el diccionario a partir del cual se creó dicho modelo de lenguaje.

Para los experimentos se han usado 9000 formularios escaneados en un proceso industrial relacionado con la tarea de introducción de datos correspondiente, al censo de España. Los campos sobre los que se ha realizado la experimentación son *provincia* (52), *municipio* (39643) que incluye entidades locales y *código postal* (11138). En el experimento se han incluido en los modelos a todas las provincias, municipios y códigos postales de España. Dichas cadenas no tenían ni probabilidades a priori ni frecuencias absolutas. Cada municipio pertenece a una provincia, la cual tiene unos pocos centenares de éstos, y cada municipio tiene uno o más códigos postales. Algunos códigos postales se comporten entre varios municipios pequeños y las grandes ciudades tienen muchos códigos postales.

El corpus se ha dividido en un conjunto de entrenamiento (10 %) y un conjunto de test (90 %). Se ha utilizado el conjunto de entrenamiento para estimar los parámetros del modelo de error (probabilidades de inserción y borrado) y los parámetros λ_h y λ_e para la operación de composición de WFST.

Tal y como se vio en la sección 4.3 los resultados que se van a mostrar a continuación muestran la tasa de reconocimiento como el porcentaje (con respecto al conjunto total de test) de las cadenas que fueron aceptadas y exitosamente corregidas por el proceso ($\frac{VP}{P+N}$), dada una tasa de error medida como el porcentaje sobre el conjunto total de test, de las cadenas que fueron aceptadas y estaban mal corregidas por el proceso ($\frac{FP}{P+N}$).

En las figuras 4.5, 4.6 y 4.7 se presenta una comparación entre los resultados de la corrección de cada campo mediante un modelo independiente y usando el modelo combinado. Cada valor en estas gráficas se deriva de la aplicación de un umbral de rechazo diferente sobre la probabilidad asociada al camino más cercano del transductor compuesto. Se puede apreciar una mejora significativa usando la aproximación de campos combinados en todos los casos.

En la tabla 4.3 se muestran las tasas de reconocimiento cuando no se rechaza ninguna cadena. Cabe destacar que estas tasas de reconocimiento corresponden a tasas de error más altas que las que se muestran en las figuras 4.5, 4.6, 4.7 y se corresponden con los límites asintóticos (final a la derecha de las gráficas) que son útiles cuando se quiere comparar el rendimiento en el peor de los casos.

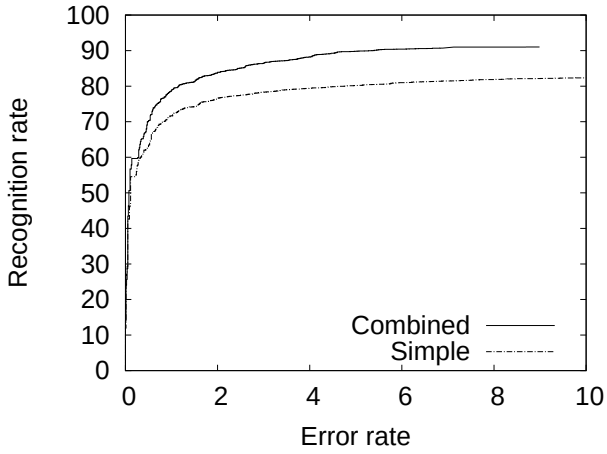

Figura 4.5: Comparativa de la tasa de reconocimiento frente a la tasa de error del campo de provincias corregido mediante un modelo simple y un modelo combinado.

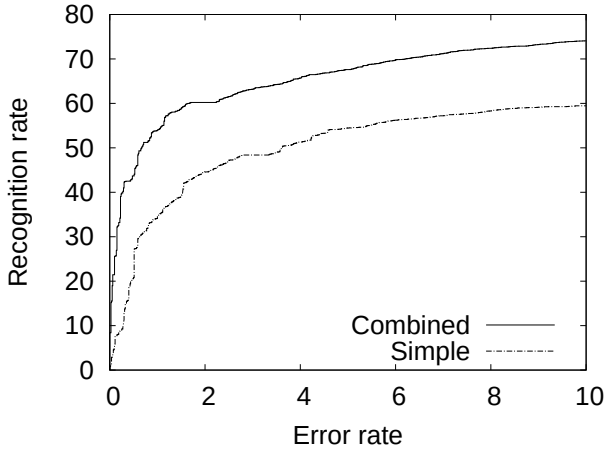

Figura 4.6: Comparativa de la tasa de reconocimiento frente a la tasa de error del campo de municipios corregido mediante un modelo simple y un modelo combinado.

	Simple	Combinado
Provincias	83,4 %	91,0 %
Municipios	64,0 %	78,5 %
Código postal	68,1 %	72,4 %

Tabla 4.3: Tasa de reconocimiento en tasa de rechazo cero.

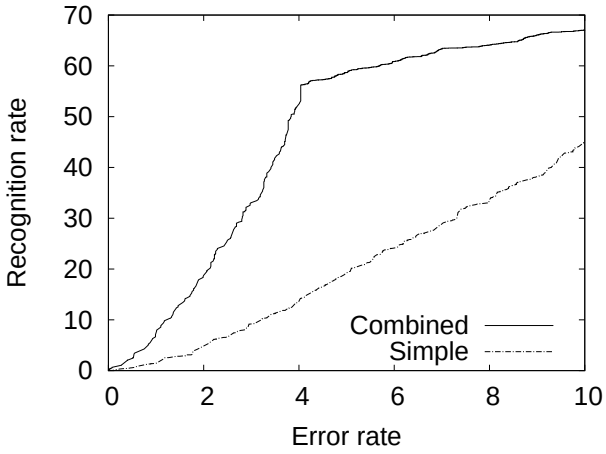

Figura 4.7: Comparativa de la tasa de reconocimiento frente a la tasa de error del campo de códigos postales corregido mediante un modelo simple y un modelo combinado.

4.4.3. Conclusiones

Se ha propuesto un método de postproceso para sistemas de digitalización de formularios manuscritos usando Transductores de Estados Finitos con Pesos (WFST) y lenguajes k-testables donde aparecen restricciones léxicas o lingüísticas de varios campos interrelacionados que han sido combinados para producir una mejora en las tasas de reconocimiento. Dicho método se basa en la concatenación de los campos haciendo uso de un símbolo especial y el establecimiento de restricciones especiales a nivel del modelo de error para este símbolo especial.

La flexibilidad ofrecida por los WFST hace que aplicar el método con este tipo de campos sea muy sencillo, pues la metodología aplicada en la composición y posterior búsqueda del mejor camino no cambia, relegando toda la complejidad en la propia representación de los autómatas a la hora de representar este nuevo tipo de campos. Los resultados han mostrado una reducción significativa en las tasas de error, gracias al incremento de contexto que el uso de los campos combinados ofrecen respecto al uso independiente de los modelos simples, sin afectar por ello a los tiempos de corrección. Ello es debido al hecho de que en los campos susceptibles de ser combinados, no todas las cadenas de un campo se pueden cruzar con todas las cadenas del otro campo, lo que hace que al combinarlos la longitud de la cadena a corregir sea mayor y por lo tanto también el número de evidencias que nos lleven a la cadena correcta dada la entrada. En los experimentos realizados sobre provincias, municipios y códigos postales la introducción de los campos combinados han producido mejoras relativas de la tasa de reconocimiento, sobre el conjunto total de cadenas, del 9.1 %, 22.7 % y 6.3 % respectivamente.

CAPÍTULO 5

ESTIMACIÓN DEL UMBRAL DE RECHAZO PARA EL POSTPROCESO OCR

Índice del capítulo

5.1. Objetivos

Un proceso OCR es seguido frecuentemente por la aplicación de una corrección mediante un modelo de restricciones, más concretamente por un modelo de lenguaje. El objetivo de esta corrección no es otro que buscar la mejor transformación, por medio de operaciones de edición (sustitución, inserción y borrado), de una hipótesis OCR en una cadena compatible con las restricciones del documento, campo o palabra bajo consideración.

El coste que supone esta transformación se puede utilizar como un valor indicativo de la confianza que se tiene en la corrección, lo que permite imponer un umbral para decidir si una cadena es aceptada o rechazada. En caso de rechazo la cadena puede ser enviada a validación humana o no, dependiendo de la tarea con la que se esté tratando. Así pues, un sistema OCR a página completa no requiere habitualmente de validación humana si su salida se utiliza, por ejemplo, para la clasificación de documentos. En este caso los *falsos positivos*, es decir, cadenas aceptadas como correctas cuando no lo son realmente, pueden considerarse más importantes que los *falsos negativos*, cadenas consideradas incorrectas que realmente si que eran correctas. Sin embargo, en otras ocasiones si que se requiere de una validación humana posterior para aquellas cadenas rechazadas por el sistema, un ejemplo de ello sería la corrección de cadenas procedentes de un formulario, donde el objetivo es obtener la información precisa que el individuo ha rellenado en dicho formulario.

En este capítulo se propone un método para la estimación automática de este umbral de rechazo que permita al usuario definir una tasa de error esperada sobre el conjunto de cadenas aceptadas (no rechazadas) de un lote completo de documentos. La aproximación se basa en la estimación de la distribución de la tasa de error esperada de las hipótesis OCR frente a su coste de transformación, para lo cual se hará uso de una muestra de hipótesis OCR supervisada. Para tareas donde no se dispone de hipótesis OCR etiquetadas para entrenar el modelo, se propone un procedimiento general para generar distorsiones en cadenas procedentes de un lenguaje y que por lo tanto ya estarán etiquetadas, al igual que en el caso anterior. Mediante este último método, tras la construcción de un nuevo modelo de lenguaje el usuario dispondrá de una muestra sintética de hipótesis OCR con la que calcular la distribución de la tasa de error esperada. En definitiva, tanto si se dispone de una muestra de hipótesis OCR real, como si no, el usuario será capaz de proponer al sistema una tasa de error que considere aceptable, y de manera transparente, el procedimiento establecerá el umbral de rechazo para un campo dado con su modelo de restricciones propio.

5.2. Introducción

A lo largo del tiempo se han empleado diferentes técnicas para postprocesar hipótesis OCR (nos referiremos como cadenas OCR por simplicidad, aunque se puede tratar de secuencias de vectores de probabilidades *a posteriori* u otras estructuras más complejas) de acuerdo con un modelo de lenguaje establecido, tanto en entornos con restricciones como sin ellas. Se pueden encontrar algunos ejemplos en Hull and Srihari [1982], Tong and Evans [1996], Perez-Cortes et al. [2000], Kolak and Resnik [2005], y Llobet et al. [2010b].

La mayor parte de ellos ofrecen, o pueden ser modificados para que así lo hagan, una estimación del esfuerzo necesario para transformar la salida del clasificador OCR de acuerdo a las restricciones establecidas por un lenguaje. Esta estimación se llama el **coste de transformación**. En algunas ocasiones, este coste se sustituye por una medida inversamente relacionada llamada **medida de confianza en la corrección** o **fiabilidad**, reflejando, de alguna manera, la probabilidad de que una hipótesis OCR pertenezca al modelo.

En el capítulo 4, se ha descrito una técnica para el postproceso de hipótesis OCR basada en transductores de estados finitos con pesos (WFST). Dicha técnica combina modelos de restricciones o de lenguaje, modelos de hipótesis y modelos de error, respectivamente, con objeto de obtener la cadena más próxima dentro de un modelo de lenguaje, teniendo en cuenta para ello, la información ofrecida por la hipótesis OCR que se obtiene del clasificador y la probabilidad de confusión de caracteres, cuya información puede encontrarse en la matriz de confusión que se puede derivar del propio clasificador OCR tal y como se comenta en el subapartado 4.3.2. Como se verá más adelante, la variable aleatoria a modelar será el coste de transformación c, que se corresponde con el valor medio de los costes que aparecen en cada una de las transiciones que representan el camino más corto en el transductor que combina todas las fuentes de información anteriormente citadas, cuyo cálculo puede observarse en la ecuación 5.1. Este cálculo es el correspondiente al de la cadena del lenguaje más probable dada la hipótesis y el resto de informaciones que tiene el citado autómata compuesto y que hemos expuesto anteriormente en el subapartado 4.3.4. Aquí trabajaremos con el valor medio de los costes, y no con el coste total, pues se pretende independizar el coste del número de operaciones de transformación, n, realizadas sobre la cadena de entrada a la hora de obtener la cadena de salida.

$$\begin{aligned} c &= \frac{\sum_{i=1}^{n} -\log P(x_i, y_i|t_i)}{n} \\ &= \frac{\sum_{i=1}^{n} -\lambda_h \log P(HM|t_i) - \lambda_e \log P(EM|t_i) - \log P(CM|t_i)}{n} \end{aligned} \tag{5.1}$$

En el trabajo preliminar Arlandis et al. [2010], se presentan resultados de la estimación de la distribución de la tasa de error esperada de un modelo de lenguaje desconocido mediante técnicas de regresión, a partir de un conjunto de entrenamiento compuesto de modelos de lenguaje conocidos. En este capítulo, se presenta una nueva aproximación que obtiene mejores resultados que los expuestos en el anterior artículo y que es más sencillo de implementar.

5.2.1. Estimación teórica del error

El control de calidad industrial puede ser abordado desde dos puntos de vista distintos. El primero de ellos toma en consideración la calidad aceptable por un receptor del producto o servicio y la calidad que el productor es capaz de ofrecer. Estas calidades se corresponden con lo que se conoce en la literatura como los riesgos del productor y el consumidor. A partir de aquí se diseñan los planes de inspección basados en el muestreo a partir de la curva

de operación característica y los riesgos comentados anteriormente. En Paladini [2000], se presenta un sistema experto para ayudar en la toma de decisiones sobre la necesidad o no de realizar una inspección y el tipo de inspección a realizar. Un segundo punto de vista se basa en el diseño de modelos que predicen la confianza de los elementos producidos a partir de una serie de variables explicativas medidas durante dicho proceso productivo. En Köksal et al. [2011] se presenta un estudio sobre diferentes modelos aplicados relacionados con la estimación de la calidad en procesos productivos.

La propuesta que aquí se presenta combina ambos paradigmas estableciendo una relación entre el *error esperado* y la variable explicativa *coste de transformación*, a partir de la cual se establece el umbral dinámico que ofrece un error final consensuado (equilibrio entre los riesgos del productor y del consumidor). El error esperado puede verse como el conocimiento que un operador experto (productor) ha adquirido a lo largo del tiempo mediante el contacto o la negociación con el consumidor del producto y puede expresarlo de manera explícita. El sistema que aquí se propone aplica un modelo de este conocimiento para convertirlo en un umbral numérico con un significado explícito poco obvio.

Para poder entender bien el significado de error en el contexto que nos ocupa, debemos introducir previamente dos conceptos que son: **1) hipótesis nula (H_0)** asociada a la presunción de que una cadena ha sido correctamente transcrita por el sistema y **2) hipótesis alternativa (H_1)** asociada a la presunción de que la cadena transcrita sea incorrecta. Fijémonos que H_0 y H_1 son dos sucesos disjuntos, por lo cual al aceptar uno de ellos como verdadero se pueden cometer dos tipos de error, tal y como viene recogido en la tabla 5.1.

Tabla 5.1: Tabla cruzada de aciertos y errores

	H_0 (real)	H_1 (real)
$\hat{H}_0$ (estimada)	Verdaderos Positivos	$\beta = P(\hat{H}_0 \mid H_1)$
$\hat{H}_1$ (estimada)	$\alpha = P(\hat{H}_1 \mid H_0)$	Verdaderos Negativos

Por un lado tenemos el error de tipo I, o riesgo de primera especie, al cual vamos a llamar α. Este error tiene asociada una probabilidad de producirse, $P(\hat{H}_1|H_0)$, que es la probabilidad de que la cadena transcrita por el sistema sea considerada incorrecta cuando realmente es correcta (falso negativo). Por otra parte tenemos un error de tipo II, o riesgo de segunda especie, al cual vamos a denominar β y que también tiene asociada una probabilidad, $P(\hat{H}_0|H_1)$, que es la probabilidad de aceptar que la cadena ha sido transcrita de manera correcta por el sistema cuando realmente no es así (falso positivo). Cada uno de estos tipos de error tendrá una importancia diferente, dependiendo de la tarea que estemos evaluando.

En la figura 5.1 tenemos una representación típica de la función de densidad de probabilidad teórica de los costes de transformación c, que es la medida que hemos decidido utilizar en el postproceso para presumir si una cadena debe ser aceptada o rechazada. Se observa que al tratarse de un coste, la distribución de dicha variable en las cadenas correctas (distribución azul) se encuentra desplazada hacia la izquierda respecto de la distribución de los costes en cadenas incorrectas (distribución naranja). Esto es así porque, por norma general, se obtienen

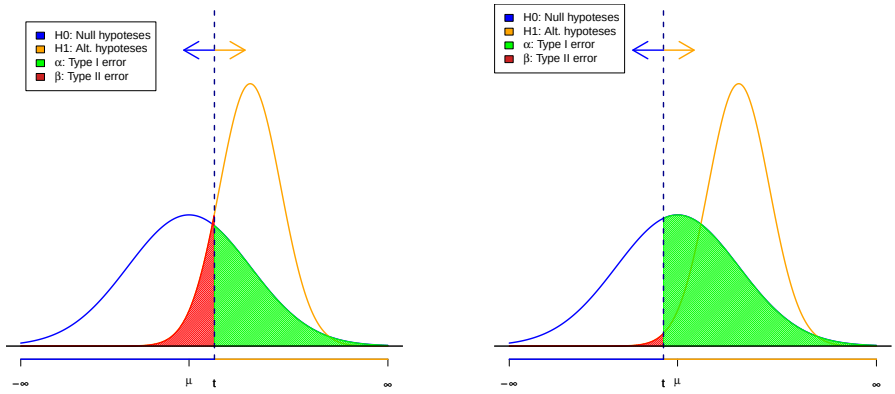

Figura 5.1: Distribución de los costes de transformación en la hipótesis H0 y H1. Dependiendo de dónde se sitúe el umbral de aceptación/rechazo t se derivarán diferentes riesgos α y β

costes más pequeños en cadenas correctas que en cadenas incorrectas debido a que si la cadena viene bastante bien reconocida a la salida del OCR entonces el esfuerzo requerido para corregirla será menor al requerir menos transformaciones que si la cadena viene peor reconocida. Por el contrario, costes mayores habrán requerido de un mayor número de transformaciones, y por lo tanto tendrán menor probabilidad de haber sido correctamente transcritas. Como se muestra en la figura 5.1, es habitual que ambas distribuciones se solapen. La magnitud de este solape, junto con el punto en el que se sitúe el umbral de aceptación/rechazo t (compárense figuras de la izquierda y la derecha), producirá diferente cantidad de errores de tipo I (α) o falsos negativos y, de tipo II (β) o falsos positivos, (anexo A.5).

La manera de proceder en un proceso como el que nos ocupa consiste en establecer un umbral, a la izquierda del cual aceptaremos de manera automática la transcripción de la cadena como correctamente transcrita y a la derecha del cual rechazaremos dicha transcripción. Así en el supuesto práctico de que sea requerida una validación manual de las cadenas rechazadas, un aumento de α=P($\hat{\text{H}}_1$|H_0) supondrá un aumento del esfuerzo de los validadores. Por otra parte tenemos el otro tipo de error, β=P($\hat{\text{H}}_0$|H_1), que está asociado al error que el usuario del sistema automático tendrá que asumir, pues en este caso aceptaremos de manera automática cadenas correctas que no lo son y que por tanto no pasarán por un proceso de validación humana, es decir, al final y suponiendo que el proceso de validación humana no cometa errores, este será el error final que se producirá. En la figura 5.1 (derecha) se observa que el umbral de aceptación/rechazo ha sido desplazado hacia la izquierda, lo que produce una clara disminución de β pero también un aumento de α, es decir, disminuimos el error final pero incrementamos el esfuerzo requerido durante el proceso de validación humana.

Si lo que se desea es disminuir ambos tipos de error, α y β, a la vez y no uno a costa del otro, entonces se plantean dos objetivos, que de manera general no tienen porque ser

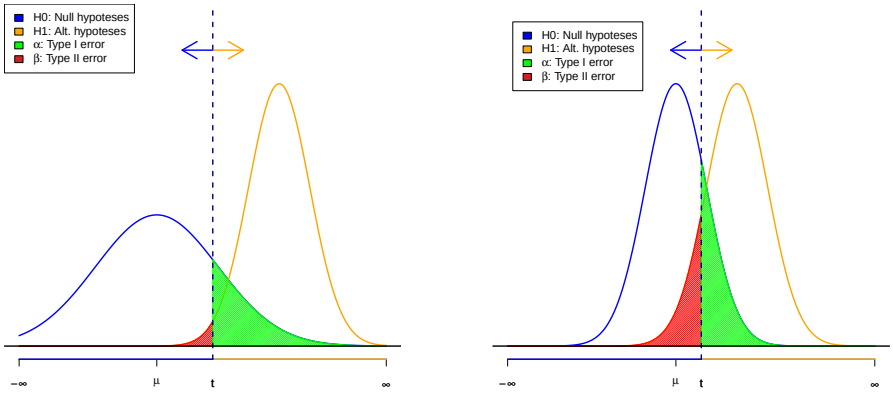

Figura 5.2: Variación en los riesgos α y β al desplazar la media de la distribución de los costes (izquierda) y al reducir la desviación de los costes(derecha), respecto de la figura 5.1(izquierda)

disjuntos, dirigidos a aumentar el grado de separabilidad entre distribuciones de diferente naturaleza. Un primer objetivo consistiría en intentar incrementar el grado de separación entre ambas distribuciones haciendo que los parámetros de posición, μ, de ambas distribuciones estén más distantes entre si, véase figura 5.2 izquierda. Por otro lado un segundo objetivo consistiría en intentar disminuir la variabilidad de las distribuciones de los costes, pues a menor dispersión menor solape entre distribuciones, véase figura 5.2 derecha. El conseguir incrementar el grado de separabilidad entre distribuciones implica mejorar las medidas de confianza o estimación de los costes mediante la introducción de nuevas variables o modelos. Este es un campo abierto y de gran interés pero no es el tema que aquí nos ocupa, por lo que se puede proponer como un posible trabajo futuro en esta temática.

A partir de estos conceptos previos, supongamos que tenemos las funciones de densidad $f_b(x)$ y $f_m(x)$, correspondientes a las densidades de probabilidad de los costes de corrección, x, de las cadenas correctamente transcritas y de las cadenas incorrectamente transcritas respectivamente. Sea además, $f_a(x)$ la función de distribución de la variable coste x, donde: $f_a(x) = f_m(x) + f_b(x)$. Se define la función de error en función del coste de corrección como: $h(x) = \frac{f_m(x)}{f_b(x)+f_m(x)}$, o lo que es lo mismo, dado un coste x, $h(x)$ nos ofrece el ratio de cadenas mal transcritas frente al conjunto total de cadenas con dicho coste. De hecho, si suponemos que tenemos un conjunto de n cadenas, cada una con su coste de transformación asociado, el error esperado en el conjunto de cadenas que no superan un umbral de coste t dado, se puede calcular como sigue:

$$
\begin{aligned}
E(h(x)|x \leq t) &= \int_{-\infty}^{t} h(x) f_a(x) dx \\
&= \int_{-\infty}^{t} \frac{f_m(x)}{f_m(x) + f_b(x)} (f_m(x) + f_b(x)) dx \\
&= \int_{-\infty}^{t} f_m(x) dx = \beta
\end{aligned} \tag{5.2}
$$

Así, si lo que se desea es no superar un *error* determinado entonces deberemos calcular dicho umbral t como sigue:

$$
\exists t : E(h(x)|x \leq t) = \int_{x=-\infty}^{t} h(x) f_a(x) dx \leq error \tag{5.3}
$$

5.2.2. Estimación del error en la transformación de cadenas procedentes de OCR

A la vista de lo anteriormente expuesto, aplicar un umbral a los costes de transformación para rechazar las hipótesis menos fiables, permite imponer un nivel de precisión sobre la salida del postproceso OCR. Nótese además que la posibilidad de regular el nivel de error esperado, en lugar de tener que tratar directamente con un umbral de coste, en una escala poco familiar y dependiente del lenguaje en cuestión, puede suponer una clara ventaja para el usuario del sistema.

Esta capacidad tiene importantes implicaciones en muchos casos prácticos. Por un lado, el usuario podría pedir al sistema una cantidad máxima de transcripciones erróneas dentro del conjunto de cadenas aceptadas (no rechazadas), lo que correspondería a una tasa de error asumible que aseguraría una mínima calidad del servicio (riesgo del consumidor). Por otra parte, en tareas donde las cadenas rechazadas son enviadas a un proceso de validación manual costoso, la cantidad de cadenas rechazadas conviene que sea tan reducido como sea posible (riesgo del productor). Todas estas consideraciones reflejan la existencia de un compromiso, en el que la selección del umbral de aceptación/rechazo juega un importante papel y puede tener un impacto significativo en el rendimiento económico del sistema.

La optimización del proceso que permite llegar a un compromiso entre ambos tipos de riesgo no es un proceso sencillo pues el número de rechazos no se puede predecir puesto que depende de la muestra particular en cuestión, siendo muy sensible a factores externos tales como la calidad de la escritura, el proceso de escaneo del documento, el realce de la imagen, el registrado de campos, etc, así como a la perplejidad del lenguaje. Por ejemplo, para un umbral determinado, la cantidad de cadenas rechazadas en dos conjuntos de documentos puede ser muy diferente si el primero de ellos esta compuesto de cadenas cuidadosamente escritas y el segundo por cadenas mal escritas, tal y como puede apreciarse en la figura 5.3.

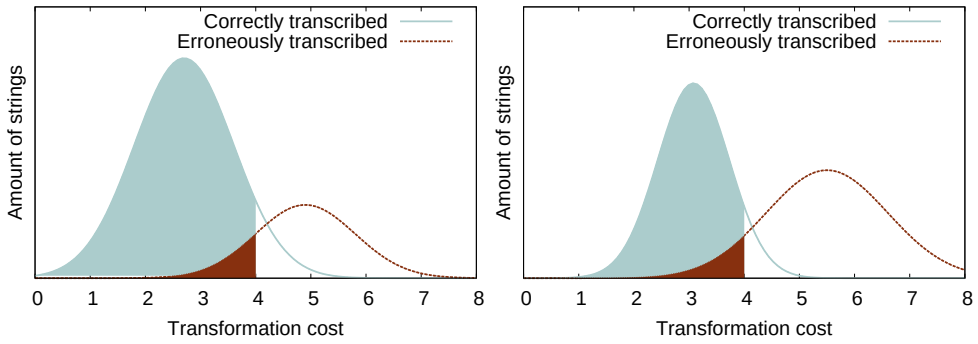

Figura 5.3: Distribución del coste de corrección entre muestras diferentes, una cuidadosamente escrita (izquierda) y otra con una escritura descuidada (derecha). Un umbral fijo llevará a diferentes tasas de error (ratio entre las áreas rellenas para el umbral $t = 4$ seleccionado) en las cadenas aceptadas en uno y otro ejemplo. Por lo tanto, se deberían aplicar diferentes umbrales para obtener la misma tasa de error en ambos casos, por ejemplo disminuyendo el valor del umbral en la muestra de la derecha.

Sin embargo, en ambos casos, la tasa de error puede controlarse haciendo uso de los umbrales de rechazo apropiados, aplicándose diferentes umbrales para los costes de transformación de cada muestra con el objetivo de obtener la misma tasa de error final.

Así pues, en un contexto real nos podemos encontrar con muestras procedentes de diferentes modelos de lenguaje y de diferente *calidad* de escritura. Si se toma una muestra de cadenas OCR, y se calculan sus costes de transformación usando un modelo de lenguaje, la distribución obtenida puede variar ampliamente para cada uno de los diferentes modelos, tal y como puede apreciarse en la figura 5.4 (izquierda) y también para diferentes muestras de un mismo lenguaje como se muestra en la figura 5.4 (derecha). Esto significa que la elección de un umbral de rechazo consistente es una tarea difícil, puesto que el número de cadenas aceptadas/rechazadas para un valor de umbral dado, variará dependiendo tanto de las características de los modelos de lenguaje como de la *calidad* en las muestras de test procesadas. De hecho, una pequeña variación en el valor del umbral puede conducir a cambios impredecibles de las ratios de cadenas aceptadas/rechazadas, así como de la tasa de error.

Una aproximación al desafío de encontrar el umbral de rechazo óptimo, consiste en usar las típicas relaciones entre umbral y error obtenido. Algunos ejemplos de este tipo los podemos encontrar en *Receiver Operation Characteristic Curve* (ROC) Fawcett [2006], la curva *Precision-Recall* (P-R) Rijsbergen [1979], y la curva *Error-Reject* (E-R) Chow [1970]. Todas estas curvas se calculan a partir de muestras supervisadas y representan información útil que suele utilizarse habitualmente para analizar y comparar rendimientos de clasificadores. Sin embargo, en las curvas ROC, P-R y E-R la precisión en las predicciones está fuertemente condicionada por la distribución de los valores de confianza encontrados en la muestra de entrenamiento, por lo que el uso de un umbral del rechazo fijo está condicionado por la naturaleza de dichas muestras. Esto implica que para obtener una misma tasa de error sobre una muestra de test ésta debería presentar una distribución de costes similar a la encontrada en

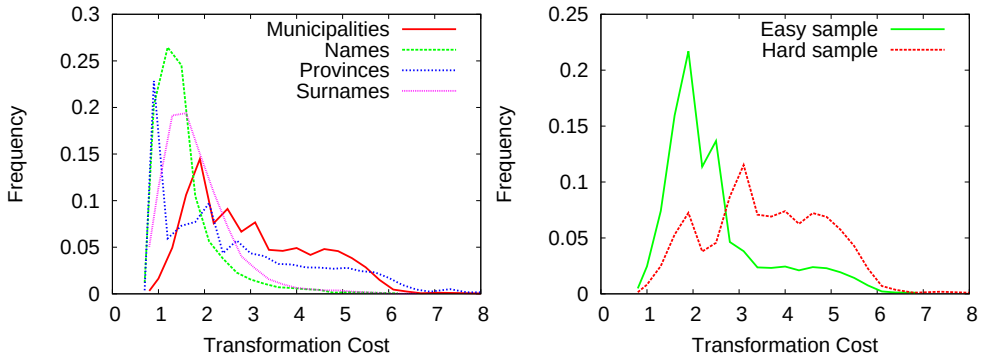

Figura 5.4: Distribución de los costes de transformación de las cadenas pertenecientes a: (izquierda) muestras de test de los modelos de lenguaje descritos en la tabla 5.2, y (derecha) una muestra de campos cuidadosamente escritos (*Easy Test*) y una muestra de campos con escritura descuidada (*Hard Test*) correspondientes a municipios (*Municipalities*). La composición de dichas muestras se detalla en el apartado 5.6.1.

la muestra de entrenamiento. Sin embargo, esta suposición es frecuentemente incorrecta. Por ejemplo, en una tarea OCR, tal y como se ha explicado anteriormente, la cantidad de errores en el reconocimiento de símbolos procedentes de un clasificador puede variar mucho para diferentes muestras, dependiendo de muchos factores, y consecuentemente, la distribución de los costes de transformación del postproceso también. Por lo tanto, basándose en la información proveniente de las relaciones mencionadas, no se debería aplicar un umbral fijo a diferentes muestras para obtener una misma tasa de error y como consecuencia, este tipo de umbralización no puede ser utilizado directamente para alcanzar nuestro objetivo.

En conclusión, se parte de la hipótesis de que la probabilidad de error de una transcripción se puede estimar a partir del coste de transformación obtenido tras la aplicación de un modelo de lenguaje (véase ecuación 5.1), y de que la predicción de la tasa de error de una muestra de test completa debe ser independiente de su distribución de costes. Se propone así una aproximación basada en umbrales de rechazo dinámicos y se aplica en dos escenarios diferentes, dependiendo de si el modelo de lenguaje utilizado es conocido, es decir, del que se tienen muestras de hipótesis OCR reales supervisadas, o por el contrario se trata de un modelo de lenguaje "nuevo" del que no se tienen todavía dichas muestras.

- **Modelo de lenguaje "conocido":** Se estima la distribución del error frente al coste de transformación, a la que llamaremos Función de Error vs. Coste, usando para esto una muestra OCR real supervisada. Las probabilidades de error asociadas a cada coste se usarán como base para la predicción de la tasa de error. Esto se describe con detalle en el apartado 5.4.
- **Modelo de lenguaje "nuevo":** La Función de Error vs. Coste de un nuevo modelo de lenguaje se estima a partir de una muestra sintética de hipótesis OCR obtenidas mediante un método de generación de errores OCR sintéticos a partir de la muestra

positiva que consiste en el léxico del lenguaje en cuestión. En este caso, al contrario que el anterior, no se requiere supervisión de la muestra lo que puede considerarse una ventaja en términos de ahorro de tiempo consumido durante el proceso de reconocimiento óptico y etiquetado manual de una cantidad significativa de cadenas. Esto es especialmente importante en la práctica, cuando es frecuente la utilización de lenguajes nuevos (incluso aunque dichos lenguajes nuevos sean subconjuntos o variantes particulares de modelos de lenguaje ya conocidos previamente). También es importante en el caso de tareas donde el etiquetado de cadenas no es posible. Este método se expone detalladamente en el apartado 5.7.

En el apartado 5.5 se presenta el procedimiento mediante el cual se utiliza la curva Error vs. Coste de transformación de un modelo de lenguaje (obtenida por medio de alguna de las alternativas expuestas anteriormente) para encontrar, de manera automática y transparente al usuario, el umbral de rechazo a aplicar sobre un conjunto de cadenas con el fin de conseguir, sobre éstas una tasa de error objetivo preestablecida por el propio usuario.

5.3. Modelos de lenguaje estudiados

Para el postprocesado de las cadenas se ha aplicado la metodología basada en WFST descrita en el apartado 4.3. Este postproceso se ha aplicado sobre las hipótesis OCR obtenidas para cuatro campos diferentes que proceden de formularios con contenidos manuscritos. Cada uno de estos campos se corresponde con un modelo de lenguaje determinista diferente donde se conoce el léxico completo de las cadenas válidas. En el estudio presentado se han postprocesado *Nombres* y *Apellidos*, simples y compuestos, es decir formados por una o varias palabras, aunque con un predominio de los simples. Estos nombres y apellidos proceden del censo de España y contienen frecuencias derivadas de sus frecuencias de aparición. Los otros modelos de lenguaje incluidos en el estudio son *Municipios* y *Provincias* del estado español. Todos estos lenguajes han sido elegidos por su representatividad en tareas reales de digitalización de formularios comerciales, ya que son muy frecuentes y abarcan un amplio rango de tallas (número de cadenas) y complejidades. Todas las muestras están supervisadas y se han obtenido a partir de un flujo de trabajo real durante el proceso de validación usando el mismo clasificador OCR.

Los modelos de restricciones o de lenguaje se han construido a partir de un algoritmo de inferencia gramatical a partir del cual se ha construido un WFST que acepta el lenguaje *k-testable* más pequeño en el sentido estricto (k-TS) consistente en un aceptor de todas las cadenas del lenguaje (Garcia and Vidal [1990]). El conjunto de cadenas aceptadas por este autómata es equivalente al modelo de lenguaje obtenido haciendo uso de $n-$gramas, para $n = k$. La extensión estocástica del lenguaje k-TS básico se realiza a través de la estimación de la máxima probabilidad de utilización de cada camino, evaluada ésta de acuerdo a la frecuencia de utilización de estos caminos en el árbol por parte de las cadenas del lenguaje. Para obtener los modelos de lenguaje deterministas se utiliza un valor de k igual a la longitud de la cadena más larga en cada uno de los lenguajes. Una vez obtenido este modelo de

lenguaje o de restricciones (CM) en forma de transductor, se compone con otros dos WFST que representan al modelo de error (EM), representando aquí a la matriz de confusión entre símbolos (incluyendo los pesos de inserción y borrado de símbolos) y el modelo de hipótesis (HM) que representa las probabilidades a posteriori del clasificador. El peso en cada una de las transiciones del autómata WFST final se obtiene a partir del logaritmo negativo de la probabilidad de transición que resulta del producto de probabilidades obtenida tras aplicar la operación de composición de WFST (calculado como la suma de logaritmos). El camino con menor coste en dicho transductor, resultado de la composición de todas las fuentes de información expuestas anteriormente, nos muestra la cadena más probable en el modelo, y la media de los pesos de las transiciones se utiliza como el coste de transformación. Este coste de transformación será la variable aleatoria que queremos relacionar con el error producido.

Tabla 5.2: Tamaño en número de cadenas de los lenguajes (muestras positivas), muestras de test, y tasas de error al 0 % de rechazo. Se muestra también el número medio de caracteres por cadena en cada uno de los lenguajes ("Long. media"). La columna "Freq." indica si el lenguaje ha sido creado con frecuencias (probabilidad de ocurrencia de cada cadena) o no (todas las cadenas son consideradas equiprobables).

Lenguaje	Tam. Leng.	Long. media	Frec.	Tam. Test	Error Test(%)
Nombres	66363	10.78	Sí	5630	2.73
Apellidos	97157	7.49	Sí	12100	4.42
Municipios	8201	11.67	No	8280	19.81
Provincias	52	7.38	No	8400	11.64

En la tabla 5.2 se muestran las características más relevantes de los lenguajes con los que se ha trabajado y sus conjuntos de muestras. Como era de esperar, el algoritmo de corrección de cadenas, alcanza tasas de error menores cuando el modelo de lenguaje ha sido inferido a partir de muestras con frecuencias, debido a que estas frecuencias son una información adicional de gran valor durante el proceso de corrección. La figura 5.4 (izquierda) muestra la distribución de los costes de transformación de las cadenas que pertenecen a las muestras de test descritas en la tabla 5.2.

5.4. Modelado de la curva Error vs. Coste de transformación.

Tal y como se ha comentado anteriormente, el resultado del postproceso a partir de una cadena obtenida mediante un modelo de clasificación OCR será una cadena corregida perteneciente al modelo de restricciones (véase el capítulo 4). Además de esta cadena, se dispondrá también de su coste de transformación, obtenido durante dicho proceso de transcripción cuyo cálculo se muestra en la ecuación 5.1. La cadena obtenida puede ser correcta o incorrecta, dependiendo de si se corresponde o no con la cadena original que estaba escrita en el formulario (representada por su etiqueta). Además, parece lógico pensar que el error asociado a una cadena postprocesada estará directamente relacionado con su coste de transformación, de

tal manera que si se ha obtenido una cadena de salida realizando operaciones de transformación poco costosas sobre la cadena de entrada, entonces la probabilidad de que sea incorrecta será más baja que si se ha obtenido a partir de operaciones de transformación con un coste elevado.

Así pues, dado un modelo de lenguaje y un conjunto de costes de transformación obtenidos al aplicar un algoritmo de postproceso a una muestra etiquetada de hipótesis OCR (muestra supervisada), se puede calcular la curva suavizada de la tasa de error en función del coste de transformación c a partir de la ecuación:

$$H(c,w) = \frac{|S^{-}_{c,w}|}{|S_{c,w}|} \tag{5.4}$$

donde w es un parámetro de suavizado referente al tamaño de la ventana utilizada (por ejemplo, rectangular de lado w), $|S^{-}_{c,w}|$ es el número de cadenas incorrectamente transcritas que tengan un coste comprendido entre $c - w$ y $c + w$, y finalmente $|S_{c,w}|$ es el número total de cadenas, correctas e incorrectas, con un coste comprendido en dicho intervalo. Aquí se podrían utilizar funciones de suavizado de ventana basadas en media móvil más complejas.

La ratio propuesta en la ecuación 5.4 se puede interpretar como la probabilidad de que un coste de transformación c provenga de una cadena incorrectamente transcrita. Para un modelo de lenguaje dado se puede utilizar H, al que vamos a denominar como *Función de Error vs. Coste (EC)*, como fuente de información a la hora de decidir el umbral del coste de transformación apropiado cuando se quiere imponer una tasa de error a una muestra dada, tal y como se explica en la sección 5.5.

Para ello, si se dispone de una muestra supervisada de cadenas postprocesadas pertenecientes a un lenguaje, entonces se puede hacer uso de dichas cadenas para el cálculo de la función EC.

En la gráfica izquierda de la figura 5.5 vemos dibujada la curva EC de cada una de las muestras de test de los modelos descritos en la tabla 5.2, usando para ello, una anchura de ventana $w = 0.25$, y utilizando las transcripciones manuales correspondientes a estas cadenas. La figura nos muestra que la probabilidad de que una hipótesis OCR sea incorrecta para un coste dado puede llegar a ser muy diferente para cada uno de los modelos de lenguaje estudiados. Como consecuencia de esto, para una tasa de error dada, se deberán de establecer diferentes umbrales para muestras de test de diferentes modelos de lenguaje.

En la gráfica derecha de la figura 5.5 se puede observar que la curva EC procedente de diferentes muestras de un mismo lenguaje es muy similar, aunque la distribución de sus costes de transformación sea diferente (según se muestra en la figura 5.4 derecha). Esto sugiere que las tasas de error en ambos casos pueden ser controladas usando una misma función EC, y por tanto que una función EC obtenida a partir de una muestra supervisada de un lenguaje puede ser asumida como modelo representativo por una muestra no supervisada del mismo lenguaje.

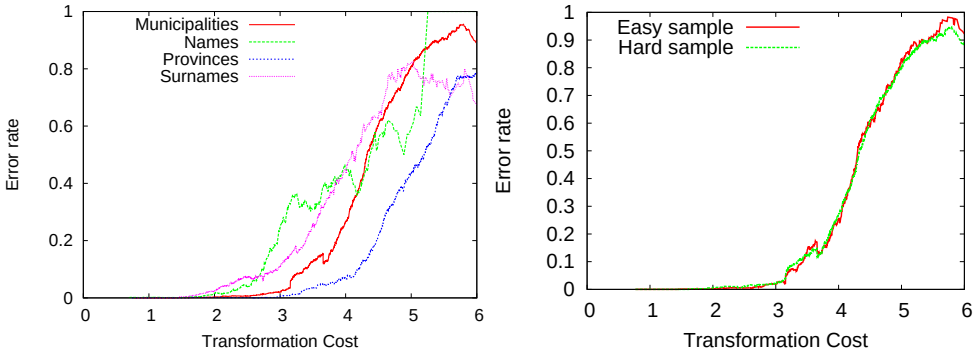

Figura 5.5: En la figura de la izquierda se observa las curvas EC de las muestras de los modelos de lenguaje descritos en la tabla 5.2 usando $w = 0.25$ (ecuación 5.4), donde se constata que cada lenguaje obtiene una función EC diferenciada. En la figura de la derecha se ha obtenido una curva EC muy similar a partir de dos muestras de Municipios (Municipalities) con diferentes distribuciones de costes, (figura 5.4 derecha): una compuesta de cadenas provenientes de palabras cuidadosamente escritas (*Easy sample*) y la otra de cadenas provenientes de una escritura descuidada (*Hard sample*).

5.5. Estimación del umbral de rechazo adaptativo en lotes

La Función del error frente al coste (EC) representada por H en la ecuación 5.4, permite obtener una estimación directa de la probabilidad de que la transcripción de una cadena sea incorrecta basándose en su coste de transformación. Así, para una cadena de la que conocemos su coste de transformación c podemos decidir aceptarla o rechazarla en función de si la estimación $H(c, w)$ es más baja o más elevada que un umbral de error dado.

No obstante, si el objetivo consiste en controlar la tasa de error de un conjunto de observaciones, por ejemplo una muestra o lote de cadenas en lugar de una única cadena, entonces se deberá aplicar aquel valor de umbral que determine la tasa de error promedio deseada para todo el lote. Así, tal y como se comprobó en el apartado 5.2.2, dada una tasa de error objetivo, se requerirá de umbrales de rechazo diferentes para lotes diferentes: los lotes que contengan un número pequeño de errores OCR contendrán costes de transformación, en general, bajos, lo que conllevará fijar umbrales más altos que en el caso de lotes que contengan un número elevado de errores OCR. Esto implicará, por ejemplo, que una cadena con un coste dado pueda ser aceptada en el primer caso, mientras que la misma cadena sea rechazada en el segundo caso. Por ejemplo, cuando la mayoría de las cadenas en una muestra tienen una elevada probabilidad de ser correctas, el error promedio esperado puede quedar por debajo del objetivo aunque algunas de las cadenas con alta probabilidad de estar erróneamente transcritas sean aceptadas. Esto se traduce en un esfuerzo de validación más pequeño y por lo tanto un incremento directo de la productividad. Y viceversa, cuando muchas de las cadenas tienen costes muy elevados, indicativo esto, de una elevada probabilidad de tratarse de cadenas erróneamente transcritas, entonces el umbral debe garantizar que la tasa de error media no supere el

objetivo, lo que podría conllevar que algunas cadenas que, de manera individual, pudieran ser aceptadas, deban ser rechazadas.

Así, para obtener una tasa de error objetivo ϵ, es posible calcular un umbral de rechazo $\mathcal{T}$ para una muestra de test de un lenguaje determinado a partir de una versión acumulada promedio de la distribución de tasas de error H de un modelo. Dada una secuencia de costes de transformación C, correspondiente a una muestra de test de hipótesis OCR, ordenadas de acuerdo a los valores de c de manera incremental,

$$C = \{c_1 \dots c_i, c_{i+1} \dots c_n\}, c_1 \leq c_i \leq c_{i+1} \leq c_n$$

se puede calcular la estimación de la tasa de error en que se incurre aceptando el subconjunto de cadenas con costes de transformación más pequeños o iguales que c_i de la siguiente manera:

$$E(c_i) = \sum_{c=c_1}^{c_i} \frac{H(c, w)}{i} \tag{5.5}$$

Y se puede obtener el umbral de rechazo $\mathcal{T}$ asociado a la tasa de error ϵ calculando:

$$\mathcal{T}_c(C, \epsilon) = \max_{E(c_j) \leq \epsilon} (c_j) \tag{5.6}$$

El valor $\mathcal{T}$ obtenido en la expresión 5.6 es el mayor de los costes donde la curva E alcanza ϵ. Puesto que la curva puede tener mínimos locales, se ha elegido el mayor valor c_j con objeto de maximizar el número de cadenas aceptadas para un ϵ dado. A la función E la denominaremos *Función Acumulada de Error vs. Coste (CEC)*. Tal y como se detalla en el apartado 5.4, dado un modelo de lenguaje, se puede obtener su distribución de la tasa de error H (expresión 5.4) haciendo uso de una muestra supervisada.

En la figura 5.6 aparecen ejemplos de estimaciones de las tasas de error calculadas mediante la función E (ecuación 5.5). En ella se observan las curvas CEC de las muestras de dos lenguajes diferentes con su correspondiente curva de error acumulado real. Los errores reales con un coste c_i se han calculado como el número de cadenas *transcritas erróneamente* y que tienen un coste menor o igual que c_i dividido por la cantidad total de cadenas que tienen un coste en el mismo intervalo (cabe destacar que esto es equivalente a calcular E usando $H(c, 0)$ sobre la muestra). Las pequeñas diferencias que existen entre ambas curvas (desviación del error) indican que se puede alcanzar una buena estimación a lo largo de todo el rango del error.

Como se comprobará en el apartado 5.6.2 de evaluación de prestaciones, el método nos ofrece una aproximación para que la tasa de error promedio de todo el lote converja a la tasa

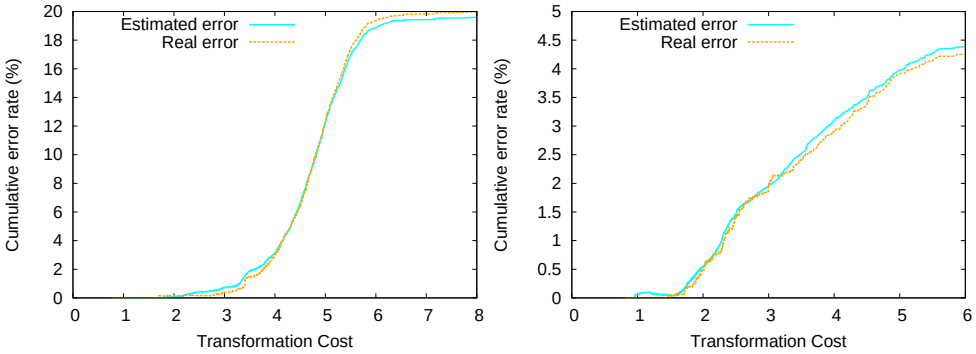

Figura 5.6: Ejemplo de curvas CEC (*Estimated error*) y error real (*Real error*) acumulado, para las muestras de Municipios (izquierda) y Apellidos (derecha) descritos en la tabla 5.2. En el caso de las curvas CEC se ha usado la mitad de la muestra disponible para calcular H mientras que la otra mitad se ha utilizado para calcular E.

de error objetivo ϵ. No obstante, en aquellos casos excepcionales en que la tasa de error objetivo sea superior al error de la muestra completa, el error máximo alcanzable será, lógicamente, el error de la muestra, siendo la tasa de rechazo del 0 %.

En algunos casos particulares, puede ser conveniente establecer un límite sobre la tasa de rechazo, y a partir de ello estimar su tasa de error correspondiente. En este caso, si las $(n - j)$ observaciones menos fiables han de ser rechazadas, la tasa de error estimada será la acumulada aceptando hasta la cadena c_j, y ésta puede ser directamente obtenida calculando $E(c_j)$. A partir de aquí, y a modo de extensión, la curva de *Error-Rechazo* tradicional se puede calcular fácilmente a partir de la curva CEC.

La figura 5.7 izquierda representa a una distribución teórica de las cadenas cuya corrección automática ha sido correcta (color azul y verde) junto a la distribución de las cadenas cuya corrección ha sido incorrecta (color naranja y rojo). El posicionamiento del umbral t delimitará, a la izquierda, el porcentaje de cadenas aceptadas como correctas, y a la derecha, el porcentaje de cadenas consideradas incorrectas. El área delimitada mediante el color rojo, a la izquierda, determinará nuestro porcentaje de error automático final, β, mientras que el área verde, a la derecha, determinará el porcentaje de cadenas, α, que aún estando bien deberán ser validadas por un humano si el proceso en cuestión así lo requiere. A partir de esta gráfica se puede apreciar que conforme más se desplace el umbral hacia la izquierda más se reducirá nuestro riesgo β y más se ampliará nuestro riesgo α y de manera inversa cuanto más desplacemos el umbral hacia la derecha mayor será nuestro riesgo β y menor será nuestro riesgo α. De manera equivalente, en la gráfica de la derecha, se observa que conforme menor sea el porcentaje de cadenas rechazadas mayor será nuestro error β.

La función CEC, conforme se ha calculado en la ecuación 5.5, se puede aplicar también incrementalmente, puesto que las cadenas de salida $\{c_1 \dots c_i \dots c_n\}$ pueden ser rechazadas o aceptadas a medida que se van produciendo por parte del sistema, dependiendo de si la tasa

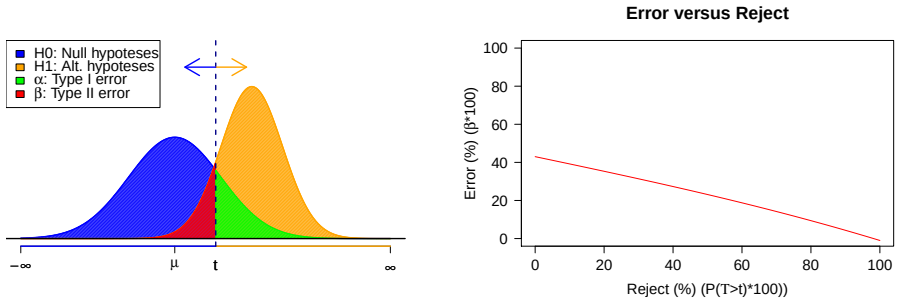

Figura 5.7: La figura de la izquierda representa las distribuciones de probabilidad de las cadenas cuya corrección automática ha sido correcta (color azul y verde) junto a la distribución de las cadenas cuya corrección ha sido incorrecta (color naranja y rojo). La gráfica de error-rechazo de la derecha nos muestra que conforme menor sea el porcentaje de cadenas rechazadas mayor será nuestro error β, mientras que cuanto mayor sea éste menor será nuestro error β.

estimada de error medio acumulada $E(c_i)$ al aceptar una nueva cadena con coste c_i exceda o no el error objetivo ϵ.

La estimación del umbral de rechazo adaptativo en lotes puede ser de gran utilidad en algunas aplicaciones prácticas, como por ejemplo, en tareas relacionadas con la digitalización de datos, donde un lote de formularios u otro tipo de documentos deben ser procesados para un cliente, que requiera una tasa de error máxima.

El método propuesto en este apartado es un método general que podría utilizarse con otros tipos de modelos de lenguaje y algoritmos de corrección de cadenas, o incluso para cualquier conjunto de observaciones si se cumple con el hecho de tener un coste de transformación (o índice de confianza) para cada observación.

5.6. Evaluación de la estimación del umbral

Los experimentos se han diseñado para evaluar la precisión del método de umbralización adaptativa presentado, véanse las ecuaciones 5.5 y 5.6, a la función de la tasa de error frente al coste de transformación (curva EC) y también pensando en la validación del procedimiento de estimación del umbral de rechazo adaptativo. Todo ello se ha realizado midiendo la habilidad que tiene el método a la hora de ofrecer un umbral que permite aproximarse a la tasa de error objetivo sobre ejemplos reales que tienen diferentes grados de representatividad respecto al corpus de entrenamiento utilizado.

Para la experimentación se han utilizado muestras OCR supervisadas, obtenidas de una tarea real de digitalización de documentos, las cuales se describen en el apartado 5.3. Con objeto de averiguar la precisión del método propuesto sobre ejemplos que provienen del mismo lenguaje pero con distribuciones de los costes de transformación distintas se han diseñado

tres conjuntos de test diferentes: *Easy Test*, formado mayoritariamente por cadenas con costes bajos, es decir, con una confianza elevada, representando una escritura cuidada; *Hard Test*, formado por cadenas con costes elevados que corresponderían a una escritura descuidada; y por último *Total Test*, que incluye los dos tipos de cadenas y representa, por tanto, un caso general en el que hay tanto cadenas fáciles como difíciles.

A las hipótesis OCR de las muestras de test de cada uno de los modelos de lenguaje descritos en la sección 5.3, se les ha aplicado el algoritmo de postproceso OCR de campos aislados explicado en el capítulo 4, obteniéndose como resultado sus costes de transformación y las transcripciones finales. Con estos datos se han calculado las funciones EC de cada uno de los modelos de lenguaje junto a las funciones CEC de cada una de sus respectivas muestras de test (*Easy, Hard* y *Total*).

Los resultados se presentan en términos de desviación del error, es decir, de diferencia entre la tasa del error objetivo ($\hat{e}_o$) y la tasa de error real (e_r), medida ésta sobre el conjunto de test. La desviación del error, $d_e = \hat{e}_o - e_r$, se presenta para valores incrementales del error objetivo. Las desviaciones obtenidas se comparan con las obtenidas aplicando un umbral fijo. Este umbral fijo supone la misma distribución de costes para cualquier muestra y por lo tanto establece el umbral de aceptación/rechazo a partir del error producido en la propia muestra de entrenamiento, cuya distribución de los costes no tiene porque corresponderse con la distribución de éstos en una nueva muestra de entrada, tal y como se observa claramente en la gráfica derecha de la figura 5.4.

A continuación en el apartado 5.6.1, se describe el diseño experimental realizado para componer los diferentes corpus que se utilizarán en la evaluación del proceso propuesto. En el apartado 5.6.2 se presentan experimentos sobre el análisis de la desviación en la estimación del error para los tipos de umbralización estudiadas, fija y adaptativa. Para ello se utiliza el corpus, *Total Test*, cuya distribución de los costes de transformación es de naturaleza similar al del entrenamiento. Para finalizar en el apartado 5.6.3 se muestran los experimentos con los corpus, *Easy Test* y *Hard Test*, que son corpus que presentan distribuciones de los costes de transformación de las cadenas muy diferentes a la del corpus de entrenamiento.

5.6.1. Diseño experimental

Para presentar resultados estadísticamente consistentes haciendo uso de las muestras de las que disponemos, aplicaremos la técnica de diseño experimental conocida como *bootstrapping*, donde para cada uno de los experimentos a realizar, se llevarán a cabo cien replicaciones. Para cada replicación la muestra de un lenguaje se divide aleatoriamente en dos mitades disjuntas: una mitad será usada para testear, mientras que la otra mitad se utilizará para el cálculo de la función EC.

A su vez, para construir los corpus de pruebas *Easy Test* y *Hard Test*, el conjunto reservado para test en cada réplica se divide en dos mitades de tal manera que: hasta el percentil 50 tendremos el saco con los costes más bajos y desde el percentil 50 hasta el final tendremos el saco con los costes más altos. Así, el *Easy Test* estará formado en un 75 % por una selección aleatoria de cadenas procedente del saco con costes bajos, mientras que el 25 % restante estará

formado por cadenas procedentes del saco con costes altos, y viceversa en el *Hard Test*. Por otra parte, el *Total Test* estará formado por la totalidad de las muestras de test. En la figura 5.4 derecha, se puede observar un ejemplo de la distribución de los costes en los test *Easy* y *Hard* para el modelo de lenguaje de Municipios (Municipalities).

A la hora de calcular la función EC con $H(c, w)$, y dado que la variable coste tiene un carácter continuo, será conveniente obtener la estimación del error para una densidad de costes mediante el uso de una ventana deslizante cuyo tamaño está relacionado con el parámetro w. La probabilidad de error para una densidad de costes se ha calculado a partir de la media móvil de una ventana rectangular centrada en el coste c y con anchura $2w$, donde $w = 0.25$. Así, para cada uno de los lenguajes en estudio, se tendrá un total de 100 funciones EC, una por replicación, obtenidas a partir de la ecuación 5.4. Además, también obtendremos la correspondiente función CEC, estimada a partir de la ecuación 5.5 para cada uno de los tres tipos de corpus de test en estudio: *Easy Test*, *Hard Test* y *Total Test*. También se ha probado con $w = 0.1$ y $w = 0.25$ siendo las estimaciones de los resultados similares, finalmente se optó por $w = 0.25$ porque las gráficas de resultados mostraban menos oscilaciones en la zona baja del objetivo.

5.6.2. Análisis de prestaciones frente a tests representativos

En el subapartado que ahora nos ocupa se muestra un análisis de la desviación en la estimación del error para los tipos de umbralización fija (*Fixed*) y adaptativa (*Adaptative*). Para ello se hace uso del corpus de test *Total Test*, cuya distribución de los costes de transformación es de naturaleza similar a la distribución de costes presentada por el corpus de entrenamiento. Para ello se calcularán los intervalos de confianza al 95 % para los dos tipos de umbralización planteados (fija y adaptativa), obteniéndose dichos intervalos, tanto para la desviación del error medio en un conjunto de lotes como para la desviación del error puntual para un único lote.

Así pues, sea la población formada por todos los posibles lotes de cadenas a postprocesar y sea la variable aleatoria $X = \{$*Desviación del error de un lote como diferencia entre el porcentaje de error estimado y el porcentaje de error real para un error objetivo dado* $(e_e - e_o)\}$.

Si $E(X) = \mu$ y $E((X - \mu)^2) = \sigma^2$, entonces si se extrae de la población de lotes, una muestra formada por N lotes, y se obtiene el estadístico media muestral ($\bar{X}$) calculado como:

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$

donde X_i es la desviación porcentual del error de uno de los N lotes para cada uno de los errores objetivo, siendo $\bar{X}$ una nueva variable aleatoria cuyo:

$$E(\bar{X}) = \mu$$

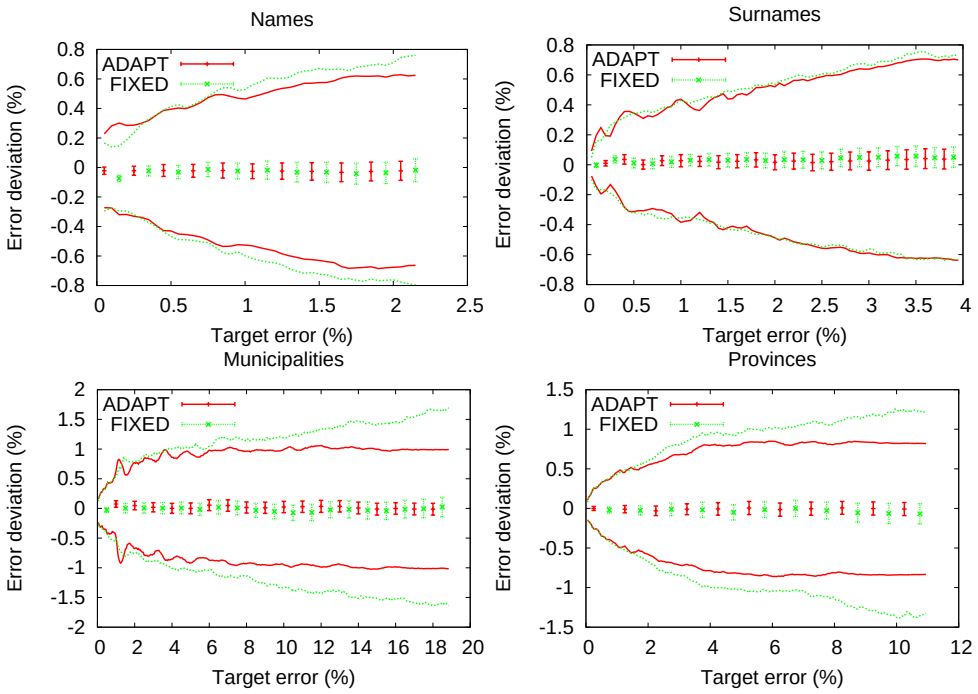

Figura 5.8: Desviación media en la estimación del error (e_e) (puntos alrededor de cero) frente al error objetivo (e_o) resultado de la aplicación de los umbrales adaptativo y fijo sobre el conjunto *Total Test*. Los valores fueron obtenidos a partir de 100 replicaciones de *bootstrapping* sobre cada uno de los lenguajes en estudio. Los intervalos de confianza al 95 % para la media muestral vienen representados mediante barras de error. También aparecen dibujados mediante líneas los intervalos $\mu \pm 2\sigma$, para una estimación puntual de un único lote.

y

$$E((\bar{X} - \mu)^2) = \frac{\sigma^2}{N}$$

A partir de estas propiedades, se puede ver que cuando se evalúa la desviación del error para una estimación puntual de un único lote, el error que se comete está relacionado con el momento central de orden 2 de la variable X, que es $E((X - \mu)^2) = \sigma^2$. Sin embargo, si la variable evaluada es una media muestral calculada a partir de N lotes, entonces su error de estimación estará relacionado con el momento central de orden 2 de dicha media muestral, que es $E((\bar{X} - \mu)^2) = \frac{\sigma^2}{N}$.

A partir de las fórmulas anteriormente expuestas, se puede observar cómo el número de evidencias (lotes en este caso) reduce el error en la estimación del parámetro poblacional μ, y es justamente esta reducción en el error lo que hace que los intervalos sean más estrechos conforme crezca el número de lotes. Además de estas propiedades, y si el número de lotes N es suficientemente grande, por el Teorema Central del Límite, $\bar{X}$ se distribuirá conforme a

una distribución Normal, independientemente de la distribución de probabilidad que tuviese la variable aleatoria original (X)[a].

Centrándonos en el análisis de los resultados obtenidos sobre el conjunto *Total Test*, la figura 5.8 muestra en los puntos centrales en torno a cero sobre el eje de las Y, la media μ de las 100 replicaciones correspondientes a valores incrementales del error objetivo (e_o) obtenidos mediante los métodos de umbralización *adaptative* y *fixed*. Además aparece también dibujado, por medio de barras de error, el intervalo de confianza (CI) al 95 %, para la media muestral, calculado a partir de una t-Student, conforme se muestra en la ecuación A.24 del anexo A.4.1, pues los datos de los que se parte son claramente muestrales y se puede asumir normalidad según el Teorema Central del Límite.

La figura 5.8 muestra que estos estadísticos son muy parecidos para todos los modelos de lenguaje y para las dos técnicas de umbralización utilizadas. Por ello, se pueden extraer las siguientes conclusiones:

- Dado que la desviación media está muy cercana a cero en todos los casos, se puede obtener una buena estimación de la tasa de error haciendo uso de cualquiera de los dos tipos de umbralización expuestos.
- Los intervalos $\mu \pm 2\sigma$ indican que la precisión en la estimación es baja cuando se utiliza un único lote. Sin embargo, los CI calculados sobre la media muestral de la desviación del error son muy estrechos lo que indica que en la práctica existirá una gran precisión en la estimación del error, incluso para errores objetivo bajos, cuando dicha estimación se haga sobre un conjunto suficiente de lotes y no sobre uno único.
- Los cuatro modelos de lenguaje probados ofrecen resultados similares a pesar de tener características diferentes, como son el tamaño del lenguaje, la tasa de error de la muestra al 0 % de rechazo, o incluso el uso de frecuencias o no en la muestra positiva del lenguaje.

5.6.3. Análisis de prestaciones frente a tests no representativos

A continuación se presenta un estudio de las desviaciones del porcentaje de error producidas mediante la aplicación de las técnicas de umbralización fija y adaptativa sobre dos conjuntos de muestras de naturaleza diferente: un primer conjunto de muestras con una calidad de escritura buena, *Easy Test*, y un conjunto de muestras con calidad de escritura mala, *Hard Test*. El objetivo de esta comparación es observar cómo afecta el tipo de umbral aplicado al porcentaje de error final estimado en muestras con distribuciones de costes diferentes, tanto entre sí, como respecto de la muestra de entrenamiento.

En la figura 5.9 se presenta la comparación entre las desviaciones obtenidas en las muestras *Easy Test* y *Hard Test* al aplicar umbralización adaptativa (*ADAPT*) y fija (*FIXED*). Se

[a] Si la variable aleatoria X ya se distribuye conforme a una distribución Normal, entonces $\bar{X}$ también se distribuirá conforme a una distribución Normal independientemente del número de muestras con las que se haya calculado.

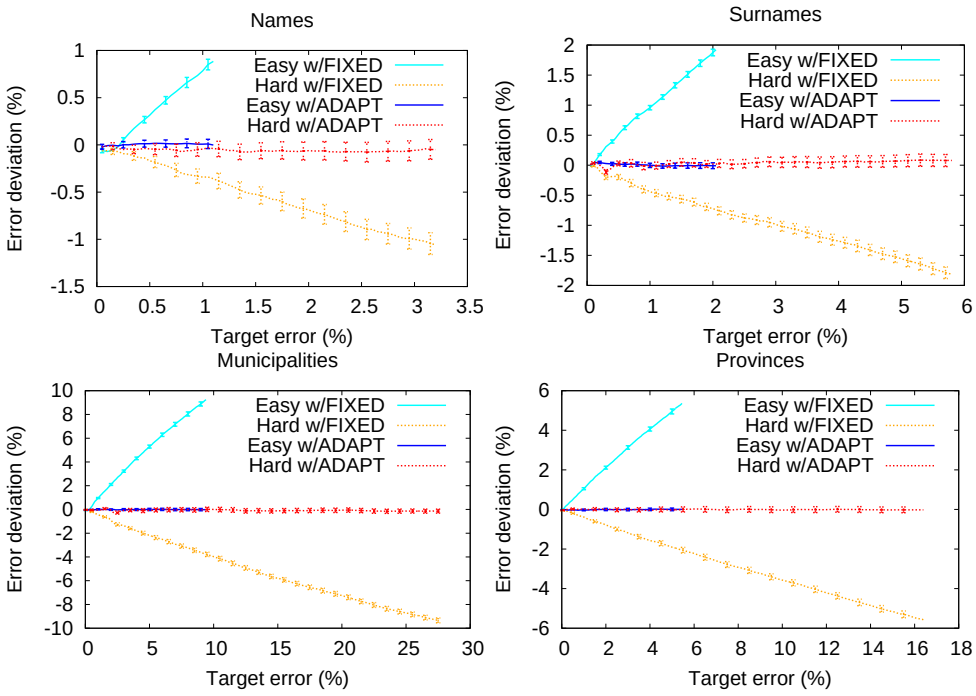

Figura 5.9: Desviación media (con intervalos de confianza al 95 %) de la estimación de la tasa de error resultante de la aplicación de los umbrales fijo y adaptativo sobre dos poblaciones que tienen distribuciones de costes diferentes: *Easy Test* y *Hard Test*. Todas las desviaciones del error se encuentran en un estrecho rango alrededor de cero cuando se hace uso de un umbral adaptativo mientras pueden llegar a ser aberrantes cuando se hace uso de un umbral fijo clásico.

muestra la media de las desviaciones del error para 100 replicaciones junto a su intervalo CI. La figura nos muestra que para todos los lenguajes y para los dos corpus de test evaluados, la media de la desviación del error tiene un rango muy estrecho alrededor del cero cuando se hace uso del umbralizado adaptativo. Esto va a ser muy útil en la práctica, pues va a permitir que el usuario decida de manera muy efectiva, qué tasa de error objetivo desea obtener para los diferentes lotes de documentos y para cualquier tasa de error.

Por contra, el método clásico de umbralización fija falla en todos los casos: en *Easy Test* las tasas de error están claramente sobrestimadas, pues el modelo predice el error por encima de la realidad, lo que nos va a llevar a un innecesario incremento de los falsos negativos, o incremento del riesgo de primera especie, α, (véase apartado 5.2), clasificando como erróneas un número elevado de cadenas que son correctas. En el caso de *Hard Test* las tasas de error se encuentran subestimadas, pues el modelo predice por debajo de la realidad, esto va a producir un incremento en el número de falsos positivos, o incremento del riesgo de segunda especie, β, (véase apartado 5.2), lo que provocaría que el sistema aceptaría automáticamente un número mayor de cadenas incorrectas que lo apropiado.

Esto nos muestra, por una parte, la falta de robustez que el método de umbralización fija clásico tiene en la estimación del error final de lotes que presentan distribuciones de costes diferentes a la utilizada durante la fase de entrenamiento, y por otra parte valida el método de umbralización dinámica presentado. Éste nos ofrece, de manera transparente, la capacidad de adaptación del umbral a la tasa de error prefijada en función del tipo de muestras a procesar en un momento dado y todo ello durante el propio flujo del trabajo de digitalización.

Tabla 5.3: Porcentaje de cadenas rechazadas como resultado de la aplicación de umbrales adaptativos, fijos y reales para un 1 % de tasa de error final, apareciendo entre paréntesis la desviación del error producida. La elevada variabilidad del rechazo entre los tres tipos de muestras de un mismo lenguaje se consigue estimar de manera muy precisa mediante umbralización dinámica, mientras que la umbralización fija causa una gran cantidad de rechazos innecesarios en la muestra *Easy Test* y una cantidad insuficiente para el caso *Hard Test*. Este comportamiento es consistente en los cuatro modelos de lenguaje probados.

	Easy Test			*Total Test*			*Hard Test*		
	Real	Adapt	Fixed	Real	Adapt	Fixed	Real	Adapt	Fixed
Names	0.80	0.80	1.93	3.99	3.96	3.91	7.88	7.48	5.89
		(0.01)	(0.78)		(-0.03)	(-0.04)		(-0.05)	(-0.34)
Surnames	3.81	3.73	10.49	20.62	20.92	20.98	46.37	46.37	31.46
		(-0.00)	(0.96)		(0.03)	(0.04)		(-0.01)	(-0.44)
Municip.	14.85	14.81	17.50	35.09	35.40	35.11	59.05	58.35	52.52
		(0.01)	(0.96)		(0.07)	(0.00)		(-0.03)	(-0.43)
Provinces	7.19	7.14	8.86	17.86	17.86	17.76	31.54	31.94	26.66
		(-0.02)	(1.05)		(-0.00)	(-0.03)		(0.02)	(-0.38)

En la tabla 5.3 aparece una comparativa sobre el porcentaje de cadenas rechazadas para una tasa de error objetivo $\epsilon = 1\,\%$ sobre cada conjunto de test haciendo uso de las estimaciones de umbralización fija y adaptativa, junto con el porcentaje de rechazo real asociado a ese mismo error del 1 %. Como consecuencia directa de los resultados mostrados en la figura 5.9, aparece un número innecesariamente elevado de rechazos cuando el tipo de umbral es *Fixed* y se procesa *Easy Test* y por otra parte aparece un número de rechazos por debajo de lo necesario, cuando se aplica el tipo de umbral *Fixed* sobre el *Hard Test*. Además, se observa una enorme variabilidad en la cantidad de rechazo dependiendo tanto del tipo de corpus como del modelo de lenguaje analizado. El control de esta variabilidad se puede considerar de enorme importancia práctica debido al impacto que esto puede acarrear desde el punto de vista de la productividad (riesgo α) y la calidad de los resultados (riesgo β). Tal y como se observa en la tabla, no es posible controlar esta variabilidad aplicando una técnica de umbralización fija clásica, mientras que sí que se consigue un ajuste fino, e independiente del tipo de muestra, haciendo uso de la técnica de umbralización adaptativa presentada.

5.7. Aproximación para modelos de lenguaje nuevos

En la práctica, cuando se define un nuevo lenguaje, es posible que no se disponga de una muestra de hipótesis OCR real para construir la curva EC de dicho lenguaje. En esta sección se propone hacer uso de las propias cadenas pertenecientes al lenguaje (muestra positiva) con el objetivo de construir un conjunto de cadenas con errores OCR generados artificialmente que sustituya a la muestra real de hipótesis. Para ello se hará uso de un modelo de generación de errores particular a partir del cual se generará la muestra sintética con la que poder estimar la curva EC de este nuevo modelo de lenguaje.

Un modelo de generación de errores OCR se define a partir del conjunto de probabilidades de inserción, borrado y sustituciones entre pares de símbolos. Haciendo uso de este modelo se pueden aplicar un número de operaciones de edición de cadenas sobre la muestra positiva del lenguaje con objeto de obtener un conjunto representativo de las nuevas cadenas. Las probabilidades de las diferentes operaciones de edición deben de ser ajustadas de acuerdo a los errores esperados, cubriendo el rango de tipos de errores reales encontrados en la salida de un clasificador OCR. El método que se propone aquí combina dos fuentes de errores diferentes: la matriz de confusión asociada al clasificador OCR, y una segunda fuente de error que se comentará más adelante y que intenta dar cuenta de otro tipo de errores.

La *matriz de confusión OCR* nos da una información valiosa, que incluye la tasa de error OCR esperada (a nivel de símbolo) y las probabilidades de confusión entre símbolos. Esta matriz se puede obtener de diversas maneras, una de ellas consiste en aplicar a la muestra de hipótesis OCR un algoritmo de alineamiento entre la cadena que representa a dicha hipótesis y su correspondiente etiqueta. Otra forma de realizar la estimación, y que puede ser utilizada en la práctica si se dispone de un clasificador OCR basado en k-vecinos, consiste en estimar la matriz de confusión OCR a partir de la salida de éste, tal y como se comenta en el apartado 4.3.2. Para ello, se utilizarán las propias muestras de entrenamiento a la hora de estimar dicha matriz, de tal manera que una vez entrenado el clasificador para cada muestra positiva se obtienen los $k+1$ vecinos más cercanos, el primero de ellos, según su proximidad, coincidirá con la propia muestra de entrenamiento es por ello que será descartado y se utilizarán las k restantes clasificaciones para realizar la estimación de ésta.

Sin embargo, hay errores que pueden aparecer en tareas reales que no son atribuibles directamente al clasificador OCR. Dichos errores proceden de fuentes de variabilidad de diferente naturaleza y normalmente son poco predecibles, provocando que algunas partes de una cadena OCR se vean seriamente afectadas. Algunos de ellos se deben a defectos durante el proceso de adquisición de la imagen o el preproceso, como por ejemplo distorsiones o traslaciones en el registrado de las imágenes, mala segmentación del carácter, borrado incompleto de las celdas que los contienen, etc. Otros tipos de errores son introducidos directamente por el propio escritor, entre los que cabe destacar tachones, faltas ortográficas, sobreescritura o caracteres poco habituales, descuidos o trazos poco usuales en el estilo de escritura, además de la inclusión de símbolos que pueden no encontrarse entre el conjunto de símbolos reconocibles por el clasificador, o también nomenclaturas alternativas como ocurre, por ejemplo, con las abreviaturas.

Las cadenas que contienen estos tipos de errores, suelen tener altos costes de transformación durante la etapa de postproceso, y no pueden ser reproducidos utilizando únicamente una matriz de confusión OCR estándar como la comentada anteriormente. Para poder generar cadenas que reflejen este tipo de variabilidad, se propone el usar una matriz de confusión adicional. Puesto que el tipo de errores que se intenta modelar aquí no siguen un patrón conocido, en este trabajo proponemos utilizar una *matriz uniforme* obtenida asignando la misma probabilidad P a todos los elementos de la diagonal (sustitución de un carácter consigo mismo), y el resto de la masa de probabilidad, $1 - P$, será distribuida uniformemente entre el resto de elementos de la fila (o columna). Estos últimos elementos se corresponden con las confusiones entre pares de símbolos diferentes. Además aquí también se pueden incluir las inserciones y los borrados si se añade para tal fin un símbolo especial.

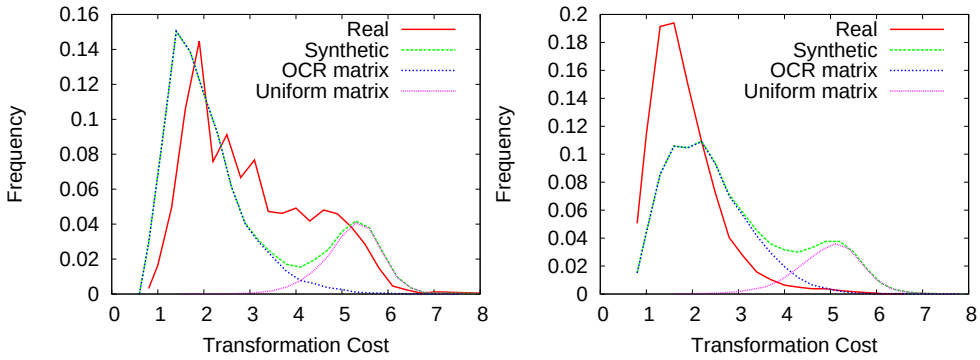

Figura 5.10: Distribución de los costes de transformación de las muestras reales y sintéticas de los modelos de lenguaje de Municipios (izquierda) y Apellidos (derecha). La curva sintética incluye a las cadenas generadas a partir de la matriz de confusión OCR y la matriz de confusión uniforme.

La figura 5.10 muestra dos ejemplos diferentes de distribuciones de costes de transformación que corresponden a cadenas generadas a partir de cada una de las dos matrices comentadas anteriormente (OCR y uniforme), además de la mixtura entre ambas (synthetic) junto con la distribución de costes de la muestra de test real. Dichas curvas muestran que las cadenas generadas por la matriz uniforme cubren un rango de valores de costes de transformación más alto, justamente en aquellas partes en las que la matriz de confusión OCR apenas aporta una pequeña proporción de las cadenas. Esto contribuirá potencialmente a una mejor estimación de la parte derecha de la curva EC.

5.7.1. Optimización de parámetros

El modelo de generación de errores propuestos requiere del ajuste de tres parámetros: número de cadenas de la muestra sintética, proporción de éstas que son generadas por las matrices OCR y uniforme, y valor de la masa de probabilidad asociada a la diagonal de la matriz uniforme. La búsqueda de los parámetros que mejor generan la muestra sintética se realizará por medio de la optimización de: a) la proporción R de cadenas generadas a

partir de la matriz de confusión OCR, siendo, $1 - R$, la proporción de cadenas generadas mediante la matriz uniforme, y b) la masa de probabilidad P que se asigna a la diagonal de la matriz uniforme. Para ello, se ha considerado la creación de una muestra sintética formada por 50.000 cadenas de cada lenguaje a partir de la distorsión de las propias cadenas del lenguaje. Se ha considerado éste como un tamaño de muestra razonable, a partir de análisis previos que buscaban el equilibrio entre el tiempo requerido para su cálculo y la obtención de una muestra suficientemente representativa.

Para la estimación de los parámetros R y P se plantea un diseño de experimentos ortogonal completo con cuatro niveles por parámetro, formado por 16 tratamientos, tal y como aparece planteado en el anexo A.3. Para la optimización de dichos factores, se propone la minimización de la función objetivo que se muestra en la ecuación 5.7, que halla la distancia entre las curvas CEC real y sintética mediante el cálculo del error absoluto entre ambas curvas, a partir de la estimación de los valores $\widehat{E}(c_i)$ y $E(c_i)$ conforme se muestra en la ecuación 5.5.

$$J(R,P) = \sum_{i=1}^{n} \left| \widehat{E}(c_i) - E(c_i) \right| \tag{5.7}$$

donde $\widehat{E}(c_i)$ y $E(c_i)$ se han calculado utilizando las curvas EC procedentes de la muestra generada de forma sintética y la muestra real *Total Test*, respectivamente, y c_n es el coste maximal entre los respectivos conjuntos sintético y real.

A partir de esta función objetivo es posible plantear un modelo, como el de la ecuación 5.8, para generar la muestra sintética. A partir de este modelo, se está en disposición de hallar los valores de los parámetros independientes R y P que minimizan la función objetivo, tal y como queda reflejado en la ecuación 5.9.

$$J = \beta_0 + \beta_1 P + \beta_2 R + \beta_3 PR + \beta_4 P^2 + \beta_5 R^2 + \varepsilon \tag{5.8}$$

$$R, P : \min J(R,P) = \min \sum_{i=1}^{n} \left| \widehat{E}(c_i) - E(c_i) \right| \tag{5.9}$$

Desde un punto de vista geométrico, para obtener una recta se requiere de un mínimo de dos puntos. Para una curva necesitamos un mínimo de tres, y para realizar un trazo curvo con punto de inflexión incluido, necesitaríamos un mínimo de cuatro puntos. Esto es lo que se correspondería con un modelo lineal, cuadrático y cúbico respectivamente. En un diseño de experimentos donde aparecen implicados diferentes factores, cada uno de los diferentes niveles sobre un factor se corresponde geométricamente con un punto en el espacio. Así pues, el número de estos niveles determinará el grado máximo del factor implicado al que se puede

llegar en la modelización de la variable objetivo en cuestión. Inicialmente, tal y como puede apreciarse en el anexo A.3, se ha planificado un diseño con los factores a cuatro niveles lo que permitiría plantear hasta un modelo de grado cúbico sobre cada uno de los factores en estudio. En el caso que nos ocupa, y para no complicar en exceso dicho modelo, se plantea hasta un grado cuadrático, como puede observarse en la recta de regresión de la ecuación 5.8. El grado final de cada uno de los parámetros se decide tras la estimación del modelo y la interpretación de la significación estadística que presenten cada uno de los parámetros tal y como se explica en el anexo A.2.

A continuación, se buscan valores comunes de los parámetros R y P que funcionen adecuadamente para todos los modelos y poder así usarlos como parámetros por omisión ante cualquier nuevo lenguaje, evitando tener que volver a estimarlos. Así pues, en lugar de estimar un modelo con los tratamientos de cada lenguaje por separado, se estima un único modelo, para todos los lenguajes, cuyas variables explicativas son R y P. Una vez obtenidos los resultados de cada tratamiento, se estima el modelo general realizándose posteriormente su optimización mediante el algoritmo de optimización presentado en el anexo A.1, con objeto de buscar los valores de R y P, que hacen mínima la función objetivo planteada.

Para la obtención de los costes de transformación de las muestras real y sintética, se ha aplicado el algoritmo de postproceso propuesto en el apartado 4.3 a cada uno de los lenguajes estudiados (Nombres, Apellidos, Municipios y Provincias). Dichos costes se han utilizado para el cálculo de las curvas EC, H y $\widehat{H}$, respectivamente (ecuación 5.4), con un tamaño de ventana, $w = 0.25$ (el mismo utilizado en todos los experimentos anteriores). Como resultado del proceso de minimización de la función objetivo, se ha llegado a unos óptimos comunes para todos los lenguajes de $R = 0.8$ y $P = 0.3$. Por un lado, esto significa que de las 50.000 cadenas que forman la muestra sintética de cada lenguaje, el 80 % de las cadenas producidas lo serán haciendo uso de la matriz de confusión OCR y el 20 % restante se producirán con la matriz uniforme. Por otro lado, esta matriz uniforme tendrá una masa de probabilidad para la diagonal de $P = 0.3$, lo que significa que una elevada masa de probabilidad, $1 - P = 0.7$, se distribuirá entre los elementos que no pertenecen a la diagonal con objeto de producir un elevado número de confusiones aleatorias entre los símbolos de la cadena sintética.

5.7.2. Evaluación de la estimación del umbral

Los experimentos en el caso de uso de muestras sintéticas, al igual que en el caso de uso de muestras supervisadas, se han diseñado para evaluar la habilidad del método de umbralización adaptativa para aproximar la estimación del error a una tasa objetivo preespecificada.

Así, utilizando los valores $R = 0.8$ y $P = 0.3$, obtenidos durante el proceso de optimización descrito en el apartado anterior, se construye una muestra sintética de talla 50000 cadenas para cada uno de los lenguajes estudiados. Tras la aplicación del algoritmo de postprocesado a cada muestra, se obtienen sus costes de transformación y su curva EC (tamaño de ventana móvil w=0.25). Finalmente, se calculan las curvas CEC correspondientes a cada uno de los conjuntos de test *Easy*, *Hard* y *Total* descritos en el apartado 5.6.1, a partir de las cuales se puede consultar el umbral asociado a las diferentes estimaciones del error objetivo.

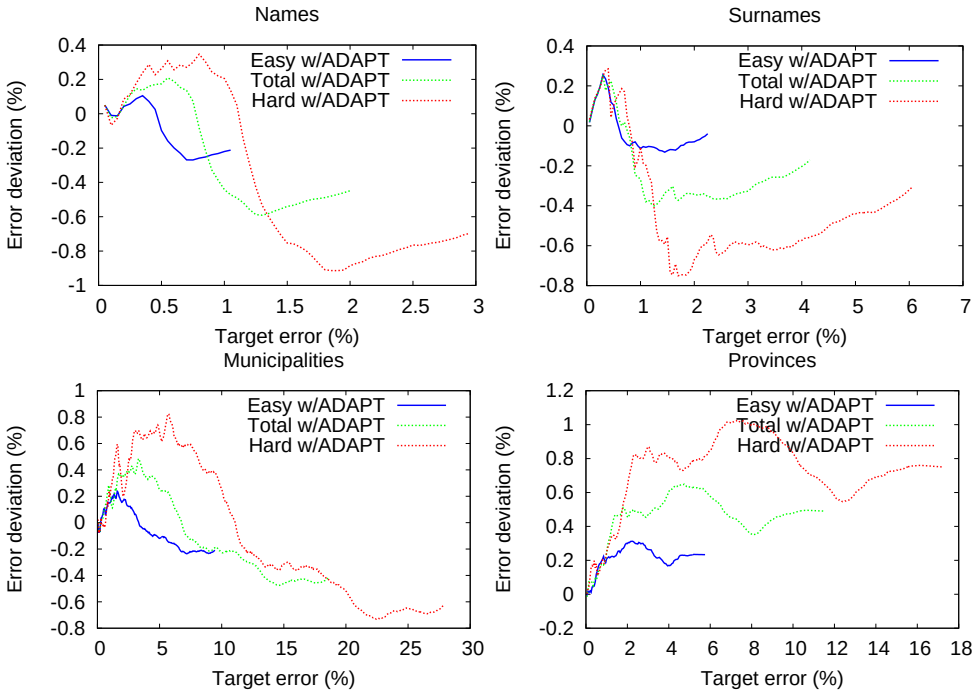

Figura 5.11: Diferencia entre la tasa de error objetivo y la tasa real (desviación media del error) calculada por medio del método de umbralización adaptativa sobre los tres conjuntos de test, *Easy Test*, *Hard Test*, y *Total Test*, diferentes distribuciones en sus costes de transformación, haciendo uso de muestras sintéticas.

La figura 5.11 muestra la media de las desviaciones obtenidas entre dicho error estimado y el error real. La desviación estándar observada (σ) y consecuentemente los intervalos de confianza derivados han sido muy similares a los obtenidos en los experimentos descritos en el apartado 5.6.2.

Considerando que no se ha utilizado información de hipótesis OCR reales de cada nuevo lenguaje, las estimaciones obtenidas para los conjuntos de prueba *Easy Test* y *Total Test* se pueden considerar útiles en un amplio rango de casos. En tareas con requerimientos de tasas de error bajas, como los sistemas de procesado de formularios, donde típicamente se aceptan tasas de error a nivel de campo que pueden estar comprendidas entre el 0.5 % y el 3 %, las desviaciones del error son suficientemente pequeñas como para ser usadas en la práctica, o como mínimo son un buen punto de partida para un proceso de refinamiento posterior. No obstante, observando el conjunto *Hard Test*, las desviaciones del error obtenidas han sido más elevadas y pueden considerarse poco precisas para algunos rangos del error objetivo en alguno de los lenguajes. En el caso de tareas con requerimientos menos estrictos de las tasas de error, como puede ser una tarea OCR para minería de datos, la estimación puede considerarse precisa y, por tanto, perfectamente útil debido a que el error relativo es muy pequeño para tasas de error medias y altas.

Tabla 5.4: Umbrales estimados y reales y porcentaje de cadenas rechazadas de acuerdo a dichos umbrales para una tasa de error objetivo del 1 % sobre las muestras fácil *Easy test* y *Hard test* haciendo uso de $\widehat{H}$ calculado a partir de una muestra sintética. En la columna *Desv* se muestra la desviación del error en valor absoluto correspondiente.

	Easy Test					Hard Test				
	Umbral		Rechazo		Desv	Umbral		Rechazo		Desv
	Estim.	Real	Estim.	Real		Estim.	Real	Estim.	Real	
Names	4.58	4.19	0.3	0.5	0.09	2.77	2.95	8.8	7.1	0.29
Surnames	3.10	2.99	3.1	3.7	0.09	2.10	1.96	41.3	50.7	0.20
Municip.	3.75	3.74	14.5	14.6	0.04	2.98	3.15	64.2	58.2	0.12
Provinces	4.66	4.82	7.3	6.9	0.29	3.90	4.10	34.5	31.0	0.27

En la tabla 5.4 se muestra el porcentaje de cadenas rechazadas en cada uno de los dos tipos diferentes de corpus, *Easy* y *Hard*, para un 1 % de tasa de error objetivo, y la comparación de umbrales de rechazo $\mathcal{T}_c$ real y estimado para cada uno de los lenguajes en ambos conjuntos de test. Tal y como era de esperar, y a la vista de los resultados, se requiere umbrales diferentes para muestras que pertenezcan a modelos de lenguaje diferentes y también entre muestras de un mismo lenguaje con distribuciones diferentes de sus costes de transformación. Además, se hace patente la gran variabilidad en la cantidad de cadenas rechazadas para los diferentes tipos de muestras dentro de un mismo lenguaje, así como entre diferentes lenguajes. Esto puede ser considerado de gran importancia práctica debido a su posible impacto económico, por lo que la aplicación de la técnica presentada en esta tesis para la estimación del error puede ser de gran utilidad tanto a la hora de ofrecer digitalizaciones con la calidad adecuada, como a la hora de optimizar la productividad del proceso.

5.7.3. Consideraciones adicionales

En la figura 5.12, se muestran las curvas EC reales (H) y estimadas ($\widehat{H}$) calculadas a partir de las muestras reales y sintéticas de los lenguajes estudiados. Las curvas EC son las mismas que se han utilizado en los experimentos del apartado anterior. Todas las estimaciones de la curva EC mostradas en las figuras han sido calculadas usando las mismas matrices de confusión y el mismo conjunto de parámetros para conseguir así resultados comparables entre los distintos lenguajes, tal y como se detalla en la sección 5.7.1.

Es importante remarcar que aquí la estimación de $\widehat{H}$ no requiere de un proceso previo de reconocimiento de caracteres o una supervisión manual final de una muestra de hipótesis OCR (pasos necesarios a la hora de obtener H), pues esto se sustituye por la muestra sintética. Esto es una ventaja importante para algunas tareas, como por ejemplo en minería de datos. Cabe destacar también que una estimación adaptativa del umbral, es estrictamente necesaria en este escenario, puesto que la construcción de una muestra sintética supone la asunción de una tasa de error intrínseca que puede llegar a ser muy diferente de la tasa de error que aparece en la muestra representativa (real), lo que conllevaría que la técnica de umbralización fija fuese completamente no usable.

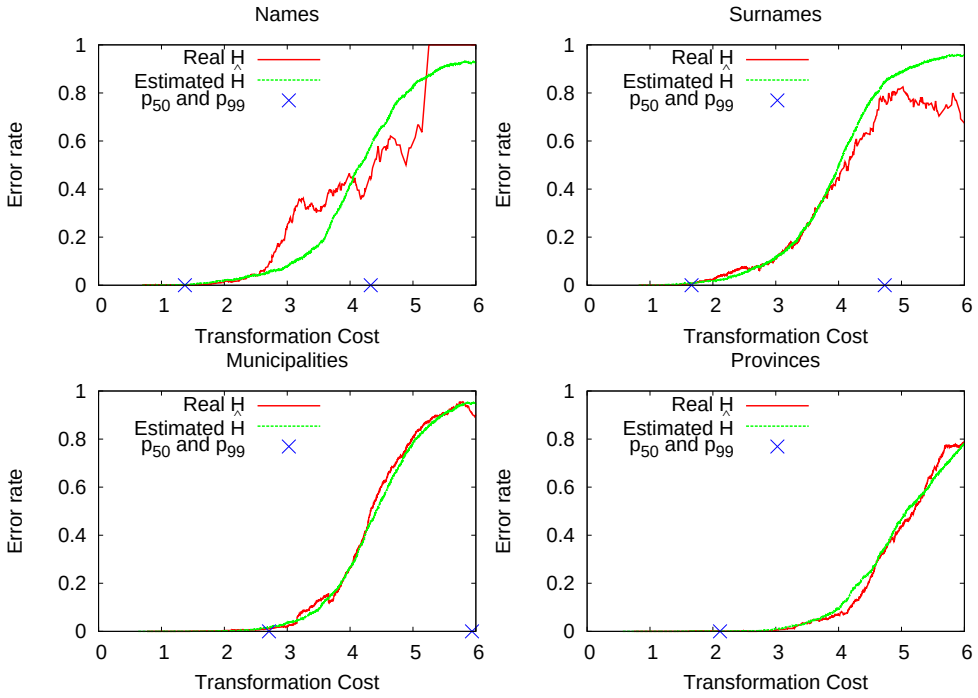

Figura 5.12: Comparativa curvas EC reales (H) y curvas EC estimadas ($\widehat{H}$) obtenidas, respectivamente, a partir de muestras reales y sintéticas. Se muestra el percentil 50 y 99 de de los costes de transformación de la muestra de test real (en el modelo de provincias, el percentil 99 se encuentra en el valor de coste de transformación = 9.2).

Cabe realizar algunas consideraciones acerca de las muestras y sus curvas EC derivadas que se presentan y analizan aquí. En la figura 5.12, se observa que las curvas EC, estimadas tienen un comportamiento más preciso para aquellos lenguajes en los que el número de muestras reales y sintéticas es más elevado respecto al tamaño del lenguaje: esta proporción es de 0.09 para Nombres (*Names*), 0.12 para Apellidos (*Surnames*), 1.0 para Municipios (*Municipalities*), y 161 para Provincias (*Provinces*), (véase tabla 5.2), mientras que la cantidad de cadenas sintéticas generadas se fija en 50.000 para todos los modelos. A partir de estos resultados, junto con la forma suave de las funciones EC estimadas ($\widehat{H}$) y que a priori sugieren la suficiente representatividad de la curva EC de los modelos estimados, cabe preguntarse si las diferencias observadas entre algunas de estas curvas en algunos de los modelos, podrían no reflejar una mala estimación del modelo sino una falta de representatividad de las muestras utilizadas como test.

En las mismas figuras se indican también los percentiles 50 y 99 de los costes de transformación de cada muestra de test real. De acuerdo con la tasa de error de cada muestra, los percentiles sugieren dos grupos de distribuciones de costes diferentes: uno procedente de

modelos de lenguaje inferidos haciendo uso de las frecuencias absolutas de aparición de cada cadena, lo cual produce costes de transformación más pequeños, como es el caso de Nombres (*Names*) y Apellidos (*Surnames*), y otro el de aquellos que son inferidos sin frecuencias con lo que todas las cadenas en ellos son equiprobables. En este último tipo de lenguajes los costes de transformación son más altos en concordancia con la mayor tasa de error al 0 % de rechazo que presentan sus muestras. El percentil 50 indica que, en todos los lenguajes, la mayor parte de los costes de transformación son bajos o muy bajos, y por lo tanto cualquier estimación considerada sobre ellas se realizará haciendo uso de la parte inicial de la cola de la izquierda de la función EC estimada ($\widehat{H}$). La distribución de costes de transformación, siguen por lo tanto, una distribución asimétrica positiva. Al mismo tiempo, los valores del percentil 99 indican que las estimaciones basadas en la parte derecha de la curva EC únicamente afectan a un conjunto muy pequeño de cadenas en el caso de Nombres (*Names*) y Apellidos (*Surnames*), los cuales fueron inferidos con frecuencias produciendo costes de transformación más bajos y por tanto tasas de error también más bajas que en Municipios (*Municipalities*) y Provincias (*Provinces*).

5.8. Conclusiones

En este capítulo se ha presentado un método y sus resultados experimentales para la estimación automática del umbral de rechazo a partir de un índice de confianza basado en los costes de transformación de una muestra de hipótesis OCR postprocesada mediante el uso de modelos de lenguaje. El objetivo consiste en que el usuario pueda decidir la tasa de error final para una muestra completa de hipótesis OCR, y el sistema estime automáticamente el umbral de rechazo correspondiente.

La relación existente entre un umbral de rechazo y una tasa de error puede establecerse mediante el uso de herramientas analíticas tradicionales, pero al utilizar este tipo de herramientas se asume que los costes de transformación siguen una distribución determinada, la de la muestra utilizada en el entrenamiento del sistema, lo que puede llegar a ser inaceptable cuando la cantidad y naturaleza del tipo de errores OCR pueden variar sustancialmente debido a la existencia de diversas fuentes de variabilidad externas. Esto conlleva cambios en la distribución de los costes de transformación, no pudiéndose aplicar así las técnicas clásicas de umbralización fija, pues esta variabilidad en la distribución de los costes no ofrece las garantías necesarias a la hora de obtener una tasa de error previamente prefijada.

Para solventar este problema se propone una técnica, de umbralización adaptativa alternativa a la de umbralización fija clásica. La técnica adaptativa presentada establece una relación entre los costes de transformación y el error producido a través de la estimación de la curva EC, a partir de los costes procedentes de una muestra preprocesada, a partir de la cual se calcula la curva CEC usando los costes de la muestra de test. Esta curva proporciona la relación final entre el error y el umbral.

Para evaluar la propuesta se han presentado los resultados experimentales sobre dos escenarios diferentes. En el primero de los escenarios la técnica se aplica a un conjunto de datos reales supervisados para obtener la relación entre la tasa de error y el coste de transformación

para un lenguaje dado. En un segundo escenario la técnica se aplica sobre un lenguaje nuevo para el cual no se dispone de muestras reales, por lo que previo a su aplicación se utiliza un método original para generar un conjunto de cadenas sintéticas con errores OCR. Estos errores son simulados a partir de la información proveniente de los errores OCR recogida en la matriz de confusión OCR junto a una matriz de confusión artificial con distribución de errores uniforme. La ventaja de este segundo escenario consiste en que no se requiere de una muestra OCR real etiquetada.

Los resultados obtenidos para el primero de los escenarios, donde el modelo es aprendido a partir de muestras reales, nos muestran que la aplicación del umbral de rechazo adaptativo produce estimaciones muy precisas de la tasa de error objetivo, con desviaciones respecto del error real por debajo del 0.1 %, tanto para un test representativo, *Total Test*, como para los conjuntos de test con distribuciones de costes muy diferentes, *Easy Test y Hard Test*. La técnica de umbralización fija en los conjuntos *Easy* y *Hard* muestra una falta muy significativa de precisión. En el segundo escenario, donde no se dispone de una muestra real, y por lo tanto es generada sintéticamente, se observa un comportamiento que puede considerarse útil de la técnica de umbralización para tareas con requerimientos de tasas de error bajos y un buen rendimiento en términos de desviación del error relativo para tareas donde no hayan requerimientos estrictos sobre la tasa de error demandada.

Parte II

Interfaces modales e interactivas

CAPÍTULO 6

MEJORAS EN LAS INTERFACES MULTIMODALES INTERACTIVAS PERSONA-MÁQUINA MEDIANTE EL USO DE WFST

Índice del capítulo

6.1. Objetivos

Tal y como quedó reflejado en el capítulo 5, las cadenas resultantes de postprocesar la salida de un motor de OCR son aceptadas como válidas en función de un umbral establecido sobre el coste de transformación a partir del error final asumido por el usuario. Dicho umbral establece la frontera entre lo correctamente postprocesado y lo que no puede considerarse como tal. Aquí se asumen dos tipos de riesgo bien diferenciados, α, asociado a FN, y β, asociado a FP (véanse tablas A.3 y A.4 del apartado A.5). Tal y como quedó patente en el anterior capítulo, el riesgo β estará directamente relacionado con el error final que el usuario es capaz de asumir, pues se corresponde con aquellas cadenas que han sido clasificadas como correctas cuando realmente no lo eran, (FP), y el riesgo α que está inversamente relacionado con la productividad durante el proceso de digitalización, pues se corresponde con el conjunto de cadenas que aún siendo correctas han sido clasificadas como no correctas, (FN), debido a que tenían un coste de transformación por encima de lo establecido. En algunas tareas, donde tan solo sea admisible un pequeño porcentaje de error, éste procederá de los FP que nunca serán revisados, pero el resto de cadenas correspondiente al grupo de cadenas rechazadas (N) si que deberán de ser revisadas, apareciendo la necesidad de una interacción humana que permita corregir y validar manualmente todo aquello que ha sido considerado como inválido tras el postproceso automático realizado. Así pues, dentro del conjunto de cadenas que superaban el umbral de aceptación, aquellas cadenas pertenecientes al riesgo α, comentado anteriormente, podrán ser validadas rápidamente simplemente aceptándolas como válidas, pero el resto de cadenas que no han superado dicho umbral requieren de un proceso costoso de transcripción. Es por ello que toda ayuda automática que se pueda ofrecer durante dicho proceso no hará más que incrementar la productividad del mismo.

En el capítulo que ahora se presenta, se muestra un método genérico de interacción simbólica de la entrada para interfaces persona-máquina que combina un modelo de hipótesis de entrada, junto a un modelo de error y un modelo de restricciones, al cual se adiciona un modelo de la interacción que el usuario va realizando durante un proceso de entrada de datos o de corrección de cadenas previamente procesadas. Para fusionar toda esta información y ofrecer una cadena postprocesada que la tome en consideración, se hace uso de los WFST que ya fueron descritos en los capítulos 2 y 3. La metodología WFST nos va a permitir fusionar la información procedente de las hipótesis iniciales, los posibles errores que pueden haberse dado durante el proceso de reconocimiento automático y las restricciones impuestas por la propia tarea además de la entrada introducida por el propio usuario en un momento dado. Todos estos modelos se combinan adecuadamente con el objetivo de obtener la cadena más probable teniendo en cuenta toda esta información en su conjunto. Para ello, inicialmente el sistema propuesto sugiere una salida basada en el conjunto de hipótesis, los posibles errores y el modelo de restricciones. A continuación, si la cadena necesita de una intervención humana, caso típico en una tarea de digitalización OCR para aquellas cadenas cuyo postproceso tiene un coste de transformación mayor al establecido, se aplica una aproximación multimodal donde la entrada de usuario se combina con los modelos anteriormente mencionados con el objetivo final de producir la salida deseada con el mínimo esfuerzo posible por parte del usuario. La aproximación que aquí se propone tiene todas las ventajas de un modelo desacoplado conservando a su vez el poder de recuperación ante errores de una aproximación integrada

donde todos los pasos del proceso se realizan en el mismo modelo (como por ejemplo un HMM en habla). Esto evita que un error en una etapa previa concreta sea irrecuperable en las siguientes etapas. En este capítulo se propone un sistema de interacción con el usuario basado en WFST. A continuación se presentan una serie de experimentos sobre dos tareas diferenciadas que van a mostrar un decremento significativo del esfuerzo de usuario conforme se va fusionando mayor cantidad de información útil.

6.2. Introducción

La excelente capacidad que los humanos tienen a la hora de interpretar un mensaje hablado, gestual o escrito, se debe mayoritariamente a su extraordinaria capacidad de recuperación de errores, gracias a las restricciones léxicas, sintácticas, semánticas, pragmáticas y discursivas que se aplican constantemente. Frecuentemente una interfaz persona-máquina está sujeta a una infinidad variable de errores e incertidumbre y la aplicación de un modelo interactivo es importante cuando el objetivo final es minimizar el esfuerzo requerido para la introducción de datos y mejorar la calidad de los datos resultantes.

En cualquier sistema de entrada de datos, desde un teclado convencional a un reconocedor OCR, incluyendo interfaces táctiles, dispositivos industriales, sistemas de navegación, control de vehículos, etc., se suelen conocer a priori los campos de entrada y elementos de diálogo, junto con la sintaxis y la semántica del propio lenguaje permitido en dichos campos. En aquellos casos en los que la forma de entrada de dichos datos sea especialmente compleja, o tenga una elevada variabilidad, como suele suceder en el reconocimiento de caracteres o de gestos, o cuando los sistemas de entrada sean poco ergonómicos debido a su tamaño, peso, factores de forma o contengan restricciones en el manejo, es cuando se requiere de manera más relevante de un método eficiente de postproceso y corrección de símbolos de entrada, con objeto de mejorar la eficiencia en la introducción de dichos datos. Otros tipos de elementos que de manera incremental van apareciendo en los sistemas más recientes, consisten en la combinación de diferentes fuentes de entrada, como pantallas táctiles, teclados, sistemas de reconocimiento de la voz, nuevos sensores como acelerómetros, GPS, etc. Así pues, cualquier sistema capaz de combinar todas las fuentes de entrada multimodales existentes va a ser de gran interés para este tipo de aplicaciones, Bastide et al. [2004], Müller and Weinberg [2011].

Formalmente y conforme se adelantó en el capítulo 1, el rol de un sistema de interacción, consiste en la maximización de la probabilidad de que las cadenas recibidas como hipótesis de entrada desde diferentes, y posiblemente multimodales, subsistemas de entrada sean correctas, en el sentido de ser compatibles con las restricciones impuestas por la propia tarea (lenguaje). Una de las principales ventajas de la aproximación que aquí se presenta radica en el uso de un modelo desacoplado, donde cada una de las diferentes partes del sistema son modelos independientes, pero que sin embargo se garantiza que la composición completa de cada una de estas partes individuales es capaz de propagar la incertidumbre de cada parte a la siguiente. En el trabajo Raman et al. [2013], se discute sobre una formulación genérica de esta idea y se presenta de una manera elegante haciendo uso del álgebra relacional y un

esquema probabilístico. En esta discusión se describen básicamente las aproximaciones desacopladas como aquellas donde cada etapa transfiere su salida a la siguiente etapa. En este caso, los errores producidos en las etapas más tempranas, debido a que cada componente realiza su búsqueda óptima local, sin tener en cuenta al resto de componentes, pueden llegar a producir una cascada de errores irrecuperables en las etapas posteriores. Sin embargo, si la incertidumbre en las predicciones se mantienen en todas y cada una de las etapas del sistema completo entonces la optimalidad global es alcanzable.

En el capítulo 4, que nace a partir de las publicaciones científicas Llobet et al. [2010a] y Llobet et al. [2010b], se propone el uso de WFST para el modelado del postproceso estocástico corrector de errores partiendo de hipótesis OCR. En estos trabajos se introdujo la fusión de diferentes modelos, incluyendo un modelo de error, y el uso de un vector de hipótesis de entrada junto con sus probabilidades a posteriori. En Perez-Cortes et al. [2011] se introduce una versión simplificada de la formulación final propuesta en el capítulo que ahora nos ocupa además de ofrecer resultados relativos a la introducción del modelo de interacción durante la propia entrada y se compara éste con un modelo de interacción básico que toma únicamente como fuentes de información la propia cadena a mejorar, el modelo de restricciones y la propia interacción, sin considerar las probabilidades a posteriori del OCR.

En el capítulo que ahora tratamos, además de ofrecer los resultados relativos a la inclusión de la interacción por parte del usuario en la tarea relativa a la mejora de cadenas procedentes de un proceso OCR, aparecen nuevas contribuciones relativas a la introducción de un nuevo modelo de error integrado dentro del propio subsistema de entrada de datos. Este último permite la recuperación de errores en el propio flujo de entrada por parte del usuario, mostrándose los resultados sobre un nuevo ejemplo de aplicación destinada a dispositivos GPS, donde además se introduce también la multimodalidad en el propio esquema propuesto. La fusión de la interactividad junto a las múltiples modalidades de entrada nos lleva a un concepto más general de *modelo de restricciones* que el utilizado hasta el momento para remplazar la noción de *modelo de lenguaje* que se citaba en los trabajos anteriores. El marco de tareas en las que se puede aplicar el paradigma aquí presentado, es mucho más amplio y general y los experimentos incluyen pruebas sobre la tarea mencionada.

El resto del capítulo se organiza como sigue: en los apartados 6.3 y 6.4 se explica en detalle el método propuesto. En el apartado 6.5 se presentan dos aplicaciones ejemplo de la implementación de dicho método: un sistema interactivo multimodal de postproceso OCR que amplia y mejora el sistema propuesto en el capítulo 4 y un método eficiente de introducción de datos orientado a dispositivos GPS. Los resultados experimentales y las conclusiones se presentan en los apartados 6.6 y 6.7.

6.3. El sistema de interacción propuesto

Se propone un sistema interactivo multimodal que combina varias fuentes de información, incluyéndose las evidencias probabilísticas, para tratar el problema de postprocesado de una hipótesis inicial que procede de un subsistema de entrada de datos, con el objetivo de obtener una salida mejorada de acuerdo a un modelo de restricciones.

Aquí se pueden identificar varias fuentes de información: a) la hipótesis de entrada, que puede ser tan simple como una secuencia de símbolos o tan compleja como un grafo representando a un conjunto de frases probabilísticas que pertenecen a una gramática. Esto es lo que nosotros denominamos modelo de hipótesis (HM), b) un modelo de errores esperados en la hipótesis de entrada junto a sus probabilidades, esto se corresponderá con el modelo de error (EM), y c) el lenguaje al cual se espera que pertenezcan las cadenas de entrada, que denominaremos modelo de restricciones (CM).

Tal y como se comentó en el capítulo 4, si todos estos modelos se combinan de la manera adecuada, el sistema propondrá una cadena compatible con CM, buscando para ello la transformación más probable de una cadena en HM a una cadena en CM a través de unas operaciones de error definidas en el propio modelo EM. Aunque en este capítulo la técnica puede ser vista como una transformación desde HM a CM, este método puede interpretarse también como una aproximación al canal de ruido, donde la cadena observada es considerada como una versión ruidosa de la cadena corregida, Brown et al. [1993], de tal manera que CM genera una cadena libre de errores con una probabilidad dada y EM, que es nuestro canal de ruido, decide si debe o no insertar errores para producir la cadena observada en el modelo HM, Park and Levy [2011].

La aproximación presentada en esta tesis es un método flexible y genérico en el que el modelo HM es capaz de codificar entradas complejas tal y como se ha mostrado en el apartado 3.2.2. Además, también permite la introducción de la información relativa a la propia interacción multimodal del usuario conforme se explicará en el apartado 6.5. Esta interacción nos es de gran utilidad en determinados casos. Un ejemplo sería cuando la cadena propuesta por el sistema de corrección no sea la adecuada y el sistema permita la interacción del usuario para la corrección de un error cometido por un proceso de corrección automático. También nos puede ser útil como fuente de información de entrada en un proceso puro de interacción con el usuario. Esta interacción tiene una naturaleza dinámica clara, pues cambia conforme el usuario va interactuando con el propio sistema y se introduce en el modelo completo a través del modelo de interacción con el usuario (IM). Este nuevo modelo será fusionado dinámicamente con el resto modelos ante cada nueva interacción de entrada que el usuario ofrezca. El objetivo a la hora de introducirlo como información es ofrecer una nueva fuente de información relevante que permita conseguir la cadena deseada con el mínimo esfuerzo.

También es posible introducir cierta tolerancia a errores durante el proceso de interacción, modelando para ello el tipo de errores que el usuario comete mientras interactúa con el propio sistema, como por ejemplo: la pulsación de una tecla adyacente a la que sería correcta, realizar una pulsación doble, etc. En este caso, el modelo IM tendrá asociado su propio modelo de errores, distinto del modelo EM, que permitirá la recuperación ante los errores cometidos por el usuario durante su interacción con el sistema. De hecho, el modelo IM puede ser visto como un segundo modelo de hipótesis, el cual nos lleva a un sistema de interacción multimodal donde está permitida la combinación de varias hipótesis de entrada a la hora de proponer una salida. Además se puede incluso concebir la coexistencia de varias fuentes de interacción, lo que incrementaría la multimodalidad del sistema. Así pues, el sistema propuesto tiene un doble propósito, primero transformar la evidencia observada (HM) en una cadena válida, basándose para ello en la aproximación de máxima probabilidad, tal y como

se ha explicado en el capítulo 3, y en segundo lugar, en el caso de que la cadena obtenida no sea la adecuada, obtenerla mediante un proceso de interacción con el usuario que implique la mínima interacción posible de éste con el sistema.

Cabe destacar aquí, que en relación a la aproximación propuesta no se ha hecho asunción alguna sobre el origen de la hipótesis inicial, o sobre la forma en la que el usuario interactúa con el propio sistema. Así pues, dicha hipótesis inicial podría provenir de una interfaz persona-máquina, como puede ser: OCR, entrada táctil, sistema de reconocimiento de gestos, sensores de entrada procedentes de un proceso físico, una secuencia biológica como ADN o cadenas de proteínas, o incluso una combinación de varios de ellos. La interacción puede realizarse a través de pantallas táctiles, teclados reducidos, o incluso a través de la voz, sistemas de reconocimiento de gestos, sistemas OCR online o cualquier sensor o conjunto de sensores diseñados para introducir acciones humanas voluntarias.

6.4. Descripción del método

El método propuesto, a la hora de abordar el problema de introducción de la interacción del usuario, va a consistir en la fusión dinámica del modelo de interacción IM con el resto de modelos utilizados durante el postproceso de las cadenas, de tal manera que cuando el usuario introduzca una nueva letra en el prefijo se creará un nuevo modelo IM que le asigne una probabilidad elevada y se volverá a buscar el camino más probable en el autómata que resulte de la composición de éste con el resto de autómatas, obteniendo así una nueva cadena resultado, conforme a la nueva restricción impuesta.

Tal y como se mencionó en el capítulo 3, se pueden identificar varias fuentes de información independientes: la hipótesis de entrada, los errores esperados y las cadenas del lenguaje a las que pertenece la tarea. Cada una de estas fuentes se puede representar por medio de un WFST a los que hemos denominado como Modelo de Hipótesis (HM) (véase apartado 3.2.2), Modelo de Error (EM) (véase apartado 3.2.3) y Modelo de Restricciones (CM) (véase apartado 3.2.1). Tal y como se vio en el capítulo 3, el autómata HM produce el conjunto de hipótesis iniciales, tras esto, el autómata EM acepta y transforma de manera expansiva y probabilística estas hipótesis iniciales a un conjunto más amplio de posibilidades y el autómata CM finalmente acepta únicamente aquellas cadenas compatibles con el conjunto de restricciones o lenguaje aceptado por la tarea de entre todas las posibles que se han presentado en su entrada.

Aunque en la mayor parte de las ocasiones es posible encontrar la cadena correcta mediante la composición apropiada de los autómatas HM, EM y CM, puede ocurrir que la cadena seleccionada en CM no fuera la que finalmente el usuario esperaba. En este caso, es necesaria la interacción con el propio usuario con el fin de obtener dicha salida final. Para modelar este proceso de interacción, se crea un nuevo WFST, al que vamos a denominar Modelo de Interacción con el Usuario (IM), véase apartado 3.2.4. Este nuevo WFST se va a encargar de modelar la información extra ofrecida por el usuario (por ejemplo un prefijo de la salida correcta) y se concibe de manera dinámica conforme el usuario va interactuando con el propio sistema, componiéndose sucesivamente con el resto de transductores. Esto impone nuevas

restricciones sobre el resto de autómatas anteriormente expuestos de manera que cuanta más información va introduciendo el usuario, más evidencias se tienen sobre la verdadera cadena y por lo tanto, el camino que representa a ésta irá ganando peso frente al resto de caminos en el autómata transductor compuesto final.

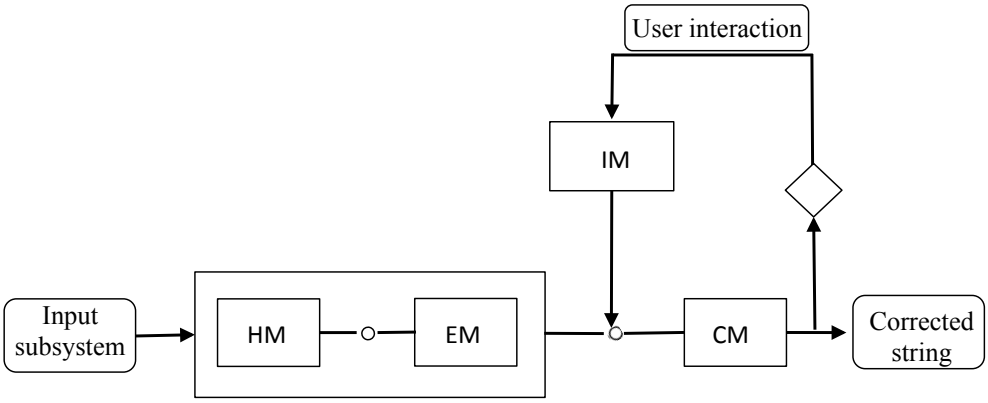

Figura 6.1: Esquema de postproceso interactivo multimodal.

La figura 6.1 nos muestra el sistema de interacción multimodal propuesto, donde el operador ∘ representa a la operación de composición de transductores. En una primera etapa, HM, EM y CM son compuestos, lo que permite la transducción de cualquier cadena $x \in L_{HM}$ en cualquier cadena $y \in L_{CM}$, de acuerdo a las operaciones de error definidas en EM. Es éste último modelo transductor, EM, el que es capaz de transducir cualquier cadena en cualquier otra. Es en la fase de decodificación donde se busca la transducción más probable de entre todas las presentes. Si la transducción propuesta por el sistema no es la que el usuario esperaba, entonces se permite la interacción de éste con el sistema para mejorar así la cadena resultante. Para ello, se crea dinámicamente un WFST llamado IM y se compone con HM∘EM, lo que impone nuevas restricciones al sistema original.

En resumen, se propone representar cada uno de los modelos (HM, EM, CM, IM) por medio de WFST, tal y como se comenta en el capítulo 3 y componerlos todos de la manera adecuada (HM∘EM∘IM∘CM) para conseguir resolver el problema de buscar la cadena más probable en CM, de acuerdo a la hipótesis actual (HM), los posibles errores del sistema generador de hipótesis (EM) y la entrada de usuario (IM), a partir de la búsqueda del camino más probable en dicho transductor compuesto. En el capítulo 3 aparece un mayor detalle, sobre cómo cada uno de estos modelos pueden codificarse de manera eficiente por medio de un WFST y sobre cómo deben de ser combinados para dar solución al problema que aquí se presenta.

6.5. Aplicaciones

Una de las principales ventajas de la metodología propuesta en esta tesis, mediante el uso de WFST y la operación de composición de autómatas es su capacidad de adaptarse a una gran variedad de aplicaciones y requerimientos. En este trabajo, se presentan y analizan dos aplicaciones diferentes como ejemplos de las posibilidades de la metodología propuesta: a) un sistema de postproceso OCR multimodal e interactivo y b) una entrada de datos eficiente en dispositivos GPS.

6.5.1. Postproceso OCR multimodal interactivo

En este apartado se muestra el ejemplo de un sistema interactivo multimodal que tiene como objeto el postproceso de un OCR. Se ha elegido como entrada el reconocimiento de caracteres en campos reconocidos procedentes de formularios manuscritos escaneados. En dicha tarea, el Modelo de Restricciones (CM) codifica la lista de cadenas válidas de un campo concreto dentro de un formulario. Si además se conoce la probabilidad a priori de cada cadena dentro de este modelo, entonces CM también codificará dicha información. Por otra parte, el Modelo de Hipótesis (HM) codifica la salida de un clasificador OCR tal y como se explica en el apartado 3.2.2. La figura 3.3 muestra el ejemplo de un transductor WFST codificando un modelo de hipótesis HM particular. En el Modelo de Error (EM) se definen las tres operaciones de edición más usuales: *sustitución*, *inserción* y *borrado*, tal y como se explica en el apartado 3.2.3 y se muestra en la figura 3.4 (izquierda).

Por otra parte, la interacción con el usuario puede ser necesaria cuando el sistema falle al buscar la transcripción correcta, este hecho puede detectarse a través de un umbral de aceptación/rechazo estimado a partir del error final asumido por el usuario tal y como se comenta en el capítulo 5. Cuando se requiera de interacción por parte del usuario para conseguir la cadena correcta, entonces se dispone de algún tipo de dispositivo como un teclado que puede ser normal, reducido, físico o virtual, o cualquier otro tipo de dispositivo de entrada de símbolos que permite al usuario la corrección de dicha cadena. Sea cual sea el dispositivo de entrada, cada símbolo introducido será visto como una *tecla pulsada*. El objetivo consiste en permitir al usuario obtener la cadena de salida correcta con el mínimo esfuerzo posible o lo que es lo mismo con el mínimo número de interacciones posibles. Para conseguir este objetivo se hace uso de un modelo de Interacción con el Usuario (IM), que codifica un prefijo tal y como se describe en el apartado 3.2.4. En esta tarea, no se ha utilizado un modelo de error (EM_{IM}) asociado a la interacción con el usuario, asumiendo con ello que la entrada del usuario está libre de errores. Cabe indicar aquí que dotar a dicha interacción de cierta tolerancia a fallos supondría únicamente modelar el tipo de errores de entrada en el modelo EM_{IM} y componerlo con el resto de modelos en el momento y lugar adecuado previo a la búsqueda de la mejor cadena, tal y como se hace en la tarea que se explica a continuación.

Los resultados experimentales de esta tarea se muestran en el apartado 6.6.1.

6.5.2. Entrada de datos eficiente en dispositivos GPS

Existen muchos dispositivos de entrada en los que la interacción con el usuario no es todo lo natural que debiera. Un ejemplo de ello se da muchas veces en dispositivos móviles o GPS donde una interacción eficiente es difícil debido al pequeño tamaño de las pantallas, o a las limitaciones inherentes de los teclados software, entre las que cabe destacar su tamaño reducido, el conjunto limitado de caracteres, o la baja precisión en la detección del puntero que en muchas ocasiones es el propio dedo del usuario.

Sin embargo, cuando se dispone de conocimiento a priori sobre la interfaz de entrada, se pueden aplicar una serie de técnicas para facilitar dicha entrada de datos. Por ejemplo, cuando se introduce una ciudad destino en un dispositivo GPS, se puede tener un conocimiento a priori de la lista con el nombre de las ciudades, el número de habitantes en dichas ciudades, los destinos más recientes, las coordenadas de localización actuales, junto al propio prefijo introducido hasta ese momento por el usuario. La fusión de toda esta información relevante acelerará el proceso de entrada reduciendo así el esfuerzo requerido en la obtención del destino deseado. Además, de manera adicional a todo esto, se puede utilizar también el conocimiento a priori de los errores que se pueden dar con mayor frecuencia durante la introducción de datos, como puede ser el pulsar una tecla adyacente a la deseada o los errores inevitables debidos a las propias deficiencias que presentan los teclados con algunos caracteres locales. Esta nueva información es la que dota al sistema de cierta tolerancia a errores en la entrada, produciendo así un sistema capaz de recuperarse frente a estos errores debidos al propio dispositivo o a defectos tipo del propio usuario.

En esta aplicación, se va a usar el método propuesto en el apartado 6.4 con objeto de reducir drásticamente el número de teclas que es necesario pulsar para introducir la ciudad o municipio de destino. Se podría utilizar también una técnica similar para introducir la dirección de destino, la provincia de destino, etc. La idea se basa en representar cada fuente de información por medio de WFST y componerlos todos juntos de la manera apropiada para conseguir que una búsqueda del camino más probable, dentro del autómata completo, nos ofrezca la mejor cadena que toma en consideración toda la información que se le haya podido aportar a través de los diferentes autómatas.

Para el desarrollo de esta aplicación, se han utilizado los modelos CM, HM, EM e IM, previamente explicados en el capítulo 3. El modelo CM es nuestro modelo de restricciones y en él se incluyen todos los municipios de España. Además, se ha utilizado la población de cada municipio para proveer de información probabilística al modelo, asumiendo así que los municipios más pobladas son las que tienen mayor probabilidad de ser seleccionadas como destino. En los experimentos presentados en el apartado 6.6.2, se ha utilizado una probabilidad proporcional a la población. El valor de k para el lenguaje k-testable, a partir del cual se genera el autómata CM (véase apartado 3.2.1), se ha establecido en el valor de la longitud de la cadena más larga, y por lo tanto se creará un transductor aceptor completo del lenguaje presentado como muestra, por lo que únicamente serán sugeridos como posibles destinos, aquellos municipios que aparecen en la propia muestra.

El modelo HM, en este caso se encarga de codificar los municipios más probables a la hora de ser elegidos como destino. Para seleccionar la lista de municipios codificados en este

modelo, junto a sus probabilidades, se ha considerado la población de éstos y la cercanía a la posición actual o a cualquier otra posición previamente elegida, bien con una simple selección sobre un mapa presentado en la pantalla, el lugar de ubicación actual, o incluso a través de la información presente en las listas que guardan los destinos más recientes. Por tanto, en este caso el subsistema de entrada que genera el modelo de hipótesis es la ubicación del GPS o ciertas coordenadas establecidas manualmente.

En nuestro caso, se ha asignado una mayor probabilidad a las ciudades más pobladas y cercanas, mediante una función que consideraba ambos parámetros bajo este criterio, asumiendo que estas ciudades tienen una probabilidad mayor de ser elegidas como destino. Por eficiencia en los experimentos presentados en el apartado 6.6.2, únicamente se han incluido en el modelo HM, las ciudades con un radio de 50 km desde la posición actual o una posición predeterminada.

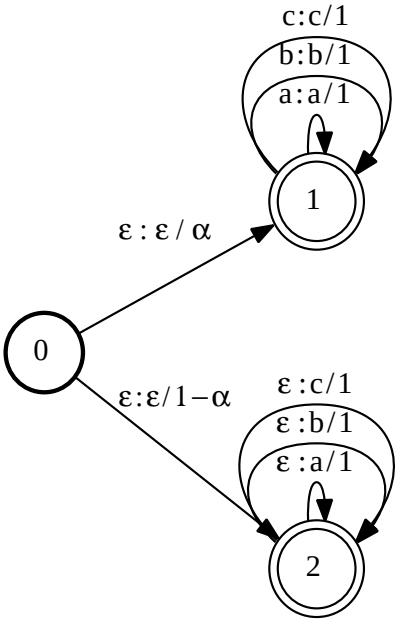

Figura 6.2: Modelo de error de HM para codificar la posible no aparición de la ciudad deseada entre las presentes en el modelo HM inicialmente propuesto en la tarea del GPS y suponiendo un alfabeto $\Sigma = \{a, b, c\}$

Es posible que el modelo HM no contenga el destino deseado, por lo tanto será necesario introducir un modelo de error EM que permita producir otras ciudades no incluidas en dicho modelo HM. Para este propósito, se ha utilizado un transductor como el de la figura 6.2 (asumiendo un alfabeto $\Sigma = \{a, b, c\}$). La rama de arriba de este transductor deja inalteradas las cadenas producidas por el modelo HM, comentado anteriormente, lo que permite que el modelo CM acepte cualquiera de estas cadenas, mientras que la rama de abajo produce cualquier cadena, lo que posibilita al modelo CM aceptar otras cadenas que no hayan sido

incluidas previamente en el modelo HM. Para conseguir que esto funcione, es necesario incluir la cadena vacía en el modelo HM, lo cual se consigue permitiendo que el estado inicial pueda ser también final. Al componer dicho modelo con el modelo EM presente en la figura 6.2, se producirá un nuevo autómata que tendrá dos partes bien diferenciadas, una que acepta las cadenas presentadas inicialmente y otra que acepta cadenas nuevas.

Para modelar la probabilidad de que el modelo de hipótesis contenga el destino deseado, se introduce una probabilidad en las ramas superior e inferior del modelo de error de HM (EM_{HM}), α y $(1-\alpha)$, respectivamente, con $\alpha \in [0-1]$. Cada una de las ramas modeladas de esta manera, pueden ser vistas como dos sucesos disjuntos. Conforme más elevado sea el valor de α más pequeña será la probabilidad de seleccionar un destino no incluido inicialmente en el modelo HM y viceversa. En los experimentos presentados se ha considerado $\alpha = 0.5$.

Finalmente, el modelo IM codifica el prefijo del nombre de la ciudad de destino según éste va siendo introducido por el usuario. Se puede dar una primera propuesta de destino combinando únicamente los modelo HM, EM y CM, sin embargo es muy probable que se requiera una selección del destino deseado. En este caso, el modelo IM se crea de forma dinámica y se compone con los modelos previos, cada vez que el usuario introduce una nueva letra al prefijo deseado y mientras éste no lo acepte como válido. Tras esto se realiza la búsqueda del camino más probable y se ofrece la cadena de salida que dicho camino codifica como resultado final, tal y como queda explicado en los apartados 3.2.4 y 3.2.5. En el caso en el que se considere que es posible la introducción de errores en la entrada, entonces se asociará al modelo IM su propio modelo de errores (EM_{IM}), pero en este caso, dicho modelo de errores estará asociado al tipo de errores relacionados con el dispositivo de entrada. Un ejemplo de este tipo de errores puede ser la probabilidad de pulsar una tecla adyacente en lugar de la correcta.

6.6. Experimentos

6.6.1. Postproceso OCR interactivo multimodal

El corpus empleado para construir los modelos de restricciones (CM) utilizados en los experimentos consiste en una muestra de 45 millones de apellidos de España con un total de 76119 diferentes. Para estudiar el efecto de la complejidad del lenguaje se han construido diferentes modelos de restricciones usando subconjuntos aleatorios de diferente tamaño al de la muestra inicial y se han seleccionado conjuntos de test específicos para cada caso, de manera que las cadenas manuscritas aparezcan siempre en el modelo de restricciones. Se ha aprendido un modelo determinista, para cada uno de los diferentes subconjuntos, para ello se ha aplicado una k igual al apellido más largo de forma que las posibles cadenas de salida son cadenas exactas de apellidos conocidos.

Como muestra de test se han utilizado 25262 apellidos manuscritos procedente de formularios escaneados en una tarea industrial real. De ellos, 14364 ($56.9\,\%$) fueron correctamente

transcritos por el sistema propuesto sin ningún tipo de intervención humana, así que para la evaluación de transcripción interactiva multimodal, que aquí se presenta, se ha hecho uso de los 10898 restantes. Es de destacar que en aquellos casos en los que es necesaria la interacción con el humano es donde la hipótesis OCR frecuentemente está muy alejada de la transcripción correcta, siendo esta distancia lo que habitualmente produce el que el sistema no sea capaz de obtener una corrección válida.

Por otra parte, el esfuerzo necesario requerido por un transcriptor a la hora de producir la transcripción correcta usando el sistema propuesto se estima mediante el ratio de teclas pulsadas *(tasa de pulsaciones de tecla (KSR))*. Este ratio se define como el número de pulsaciones sobre un teclado (u otro sistema de entrada de símbolos) que es necesario pulsar hasta alcanzar finalmente la transcripción de referencia, dividido por la longitud de la cadena.

Cuando se calcula el KSR, se debe de considerar que el usuario teclea el prefijo necesario más pequeño hasta alcanzar la transcripción deseada, esto es, el usuario comienza a escribir la cadena correcta desde el principio, aunque los primeros caracteres sean ya correctos. Este es un escenario más real, donde el usuario utiliza en todo momento el teclado y no usa el ratón para posicionar el cursor en el primer carácter incorrecto, pues hacerlo así incrementaría el tiempo de interacción. Sin embargo, puede ser interesante la distinción entre las teclas de aceptación de símbolos, como por ejemplo pulsar la flecha de la derecha o el tabulador, cuya pulsación requiere un coste temporal menor y pulsar la tecla correspondiente al *carácter correcto* en la posición de la cadena en la que nos encontramos. Es por esto por lo que el KSR se puede dividir en dos nuevas medidas que se van a llamar *tasa de pulsaciones de caracteres (CSR)* y *tasa de pulsaciones de teclas de desplazamiento (ASR)*. Así, el CSR mide el esfuerzo de pulsar nuevos caracteres y el ASR mide el esfuerzo de chequear y aceptar los caracteres propuestos, siendo la suma de ambos el KSR (KSR=CSR+ASR).

Tabla 6.1: Transcripción propuesta dada una hipótesis OCR y el prefijo introducido por el usuario

Hipótesis OCR	Prefijo	Transc. propuesta	Transc. correcta
CEIECONP	[none]	CARDONA	DELGADO
	D	DE LEON	
	DE	DE LEON	
	DEL	DELGADO	
ICERICILC	[none]	CERVELLO	RODRIGUEZ
	R	RODRIGUEZ	
BURCILI	[none]	BURRIAL	BURGOS
	B	BURRIAL	
	BU	BURRIAL	
	BUR	BURRIAL	
	BUR**G**	BURGOS	
ONAVES	[none]	NAVES	CHAVES
	C	CHAVES	
HICICA	[none]	MICLEA	MUGICA
	M	MICLEA	
	M**U**	MUGICA	

Se ha realizado una transcripción interactiva simulada, donde el usuario debe de teclear el prefijo más pequeño necesario hasta obtener la palabra correcta, que en el caso que nos ocupa coincidirá con un apellido. La tabla 6.1 nos muestra un ejemplo sobre como, dada una hipótesis OCR, el sistema propone nuevos apellidos conforme el usuario va añadiendo nuevos caracteres al prefijo. Los caracteres subrayados en negrita en el prefijo se contabilizan como CSR mientras que los otros caracteres son contabilizados como ASR.

Se ha realizado un primer conjunto de experimentos para probar la mejora alcanzada al introducir dinámicamente la interacción producida por el usuario junto al resto de fuentes de información. Con este objetivo, se han considerado cuatro combinaciones diferentes de fuentes de información (modelos): i) *Manual*, en este caso no aparece ningún tipo de modelo y el usuario debe de introducir manualmente la cadena completa; ii) *Texto Predictivo* en este caso tan solo se consideran los modelos IM y CM, asumiendo que no hay un postproceso previo ni etapa de corrección automática; iii) *Corrección de errores*, este caso representa el uso del sistema de corrección automática sin integrar la interacción con el usuario, haciendo uso del modelo HM + EM + CM; y iv) *Interacción Multimodal* en la que al modelo anteriormente mencionado se le incluye también el modelo IM, haciendo uso para ello del modelo HM + EM + CM + IM.

Se calcula el KSR, CSR y ASR en todas las aproximaciones mencionadas arriba y para los dos conjuntos de test definidos (*Total* y *Mal Corregidas*). Obviamente, la aproximación *Manual* se incluye únicamente como referencia pues en este caso el KSR es siempre 1.

La tabla 6.2 muestra el KSR, CSR y ASR para las dos primeras aproximaciones. Estos resultados muestran que el esquema de texto predictivo propuesto ofrece una reducción de 0.47 en el KSR respecto de una digitalización completamente manual en la primera de las pruebas, y de 0.42 en la segunda.

Tabla 6.2: Resultados de KSR, CSR y ASR para las aproximaciones *Manual* y de *Texto Predictivo*.

	Total			*Mal Corregidas*		
Aproximación	KSR	CSR	ASR	KSR	CSR	ASR
Manual	1	1	0	1	1	0
Texto Predictivo	0.53	0.40	0.13	0.58	0.44	0.15

La tabla 6.3 muestra el KSR, CSR y ASR para las aproximaciones *Corrección de errores* e *Interacción Multimodal* y para los conjuntos de test *Total* y *Mal Corregidas*. En este caso los resultados obtenidos, comparando la aproximación *Interacción Multimodal* frente a la de *Corrección de errores*, muestran una ganancia neta, en términos de KSR, de 0.23 en el conjunto de test *Total*, mientras que en el conjunto de test más difícil, *Mal Corregidas*, la ganancia neta llega a 0.32. Estas ganancias pueden ser vistas como el ahorro, en términos de requerimientos humanos, alcanzado cuando el modelo IM se compone junto al resto de fuentes de información.

Tabla 6.3: Resultados de KSR, CSR y ASR para las aproximaciones *Corrección de errores* e *Interacción Multimodal* sobre los conjuntos *Total* y *Mal Corregidas*.

	Total			*Mal Corregidas*		
Aproximación	KSR	CSR	ASR	KSR	CSR	ASR
Corrección de errores	0.55	0.24	0.31	0.83	0.51	0.32
Interacción Multimodal	0.32	0.23	0.09	0.51	0.33	0.18

Por otra parte, la comparación entre *Texto Predictivo* e *Interacción Multimodal* nos muestra la ganancia alcanzada cuando se incorpora dentro del esquema de interacción con el usuario una fuente de información extra, como es la salida OCR en este caso, junto con un método de corrección del error. En este caso, los resultados muestran una reducción de 0.21 y 0.07 en el KSR para los conjuntos respectivos *Total* y *Mal Corregidas*.

Se ha llevado a cabo un segundo conjunto de experimentos para comprobar la influencia del número de cadenas válidas en relación con la complejidad del modelo CM, usando para la evaluación el conjunto de test *Mal Corregidas*. Para ello, se han creado subconjuntos de cadenas válidas, extraídas aleatoriamente de las muestras de apellidos, con las que se construyen diferentes modelos de restricciones de tamaños crecientes.

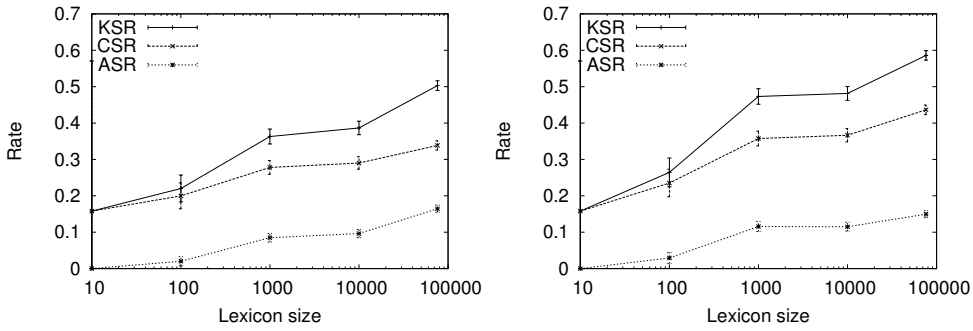

Figura 6.3: KSR, CSR y ASR junto a sus intervalos de confianza al 95 %, para diferentes tamaños de léxico, en una aproximación *multimodal* (izquierda) y su aproximación *no-multimodal* (derecha).

La figura 6.3 (izquierda) muestra los valores de KSR, CSR y ASR, junto con sus intervalos al 95 %, para tamaños incrementales del modelo de restricciones, cuando se utiliza una aproximación interactiva *multimodal*. Para poder comparar el método presentado con una aproximación *no-multimodal*, se ha predicho la entrada del usuario considerando como únicas fuentes de información la propia entrada introducida por el usuario y el modelo de lenguaje, no añadiendo ningún otro tipo de información extra y sin tener en cuenta tampoco la información aportada por la hipótesis OCR mediante el modelo HM. Se han realizado los mismos experimentos que en el caso anterior. Los valores de KSR, CSR y ASR del experimento *no-multimodal* se presentan en la figura 6.3 (derecha). Aunque nos encontramos en un caso en el que la hipótesis OCR en la muestra de test es particularmente difícil, pues se trata de cadenas en las que el sistema automático ha fallado, siendo incapaz de corregirlas

correctamente, los resultados obtenidos muestran que la aproximación *multimodal*, en la que tanto la hipótesis OCR como la entrada del usuario se consideran fuentes de información, mejoran significativamente al método *no-multimodal*.

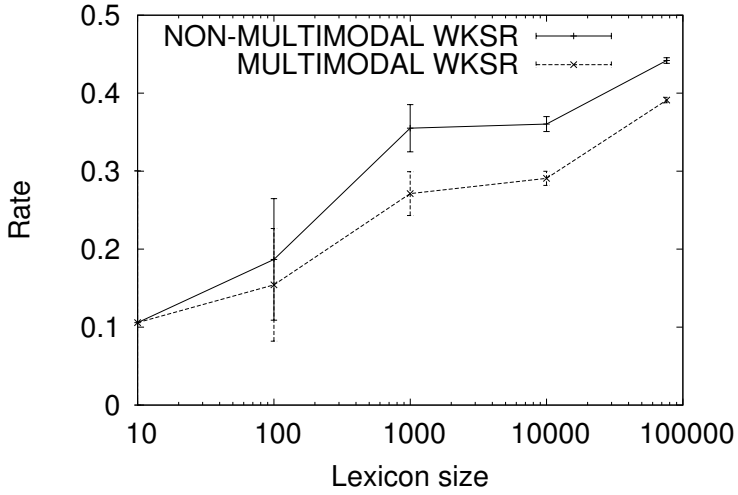

Figura 6.4: Comparación del KSR ponderado (WKSR), junto a sus intervalos de confianza al 95 %, para una aproximación *multimodal* y otra aproximación *no-multimodal*.

Si se considera que una pulsación de aceptación requiere menos esfuerzo que una pulsación de cambio de carácter, se puede utilizar un KSR ponderado (weighted key stroke ratio (WKSR)). La figura 6.4 compara un WKSR=0.66·CSR + 0.33·ASR, obtenido indistintamente en la aproximación *multimodal* y *no-multimodal*.

El coste computacional es otro tema importante en esta tarea. Se ha de considerar que en la práctica el tamaño de los modelos puede llegar a ser muy grande y que las operaciones básicas consisten en el reconocimiento de grandes lotes de documentos. Para analizar el tiempo de respuesta se obtuvo el tiempo medio de interacción cuando se disponía de un corpus grande tanto para un sistema *multimodal* como para un sistema *no-multimodal*. Lo que se encontró es un incremento significativo en el tiempo de respuesta de la versión *multimodal* respecto a la versión *no-multimodal*, pero en el peor de los casos el tiempo medio de interacción es de aproximadamente 3.0 ms el cual es mucho menor que el tiempo requerido por la interacción humana más rápida. Las medidas de tiempo han sido obtenidas sobre un procesador Intel Xeon 2.5 GHz con 2GB de memoria, con sistema operativo Linux y compilador gcc4.4. Estos resultados fueron obtenidos para un modelo de restricciones construido a partir de un vocabulario formado por 76119 palabras. El uso de modelos de restricciones más grandes no va a ser un factor excesivamente crítico pues el coste computacional crece sublinealmente y estamos ante valores temporales de respuesta muy por debajo del coste temporal de una interacción humana.

6.6.2. Introducción eficiente de datos de entrada en dispositivos de navegación GPS

Se ha utilizado el nombre de 7660 municipios de España en la construcción del modelo de restricciones. Como en experimentos previos, se ha utilizado una k igual a la cadena más larga dentro del conjunto de nombres de municipios de España, generando así un modelo determinista, que al final ofrecerá como salida alguno de los 7660 municipios de la lista.

Como la aplicación no está embebida dentro de un dispositivo de navegación GPS, para simular la introducción de datos se han utilizado teléfonos móviles con teclado táctil *qwerty*. Se han creado 84 conjuntos de datos de test, cada uno de ellos contiene 30 destinos (municipios) seleccionados aleatoriamente, lo que produce un total de 2520 destinos. Dichos conjuntos conforman nuestra muestra que fue obtenida por un conjunto de 21 usuarios que transcribieron 4 conjuntos cada uno (30x4=120 destinos por usuario).

Cada conjunto de test ha sido construido como sigue: primero se ha hecho una selección aleatoria de un municipio perteneciente a España, cuyas coordenadas de latitud y longitud serán las que representen a este conjunto. A partir de estas coordenadas, se han elegido 30 municipios utilizando un proceso aleatorio con remplazamiento que considera la proximidad de los municipios al punto seleccionado como semilla, esta cercanía se construye de manera radial. A parte de la distancia también se ha considerado la población como factor influyente sobre la probabilidad de selección del resto de municipios que formará parte de dicho conjunto de test. Así, conforme más pequeña sea la distancia a la coordenada seleccionada y más elevada sea la población, mayor será la probabilidad de que un municipio sea seleccionado para formar parte de este conjunto. Esto se ha hecho de esta manera porque se quiere medir también el rendimiento del sistema, considerando no sólo la posibilidad de buscar municipios cercanos a un punto dado, sino que también exista una posibilidad razonable de seleccionar municipios alejados de dicho punto central, situación que se suele dar comúnmente al viajar.

Para simular el proceso de corrección, comúnmente utilizado en dispositivos de navegación, se ha pedido a los usuarios que utilicen únicamente el dedo índice durante la introducción de datos. Además, también se les ha pedido que no corrijan los caracteres incorrectos introducidos durante el ingreso de caracteres, con objeto de medir así, la habilidad del sistema a la hora de recuperarse de los errores durante el proceso de interacción con el usuario (tolerancia a errores). De los 2520 destinos introducidos, 1920 (76.2 %), han estado libre de errores, mientras que el resto, 23.8 %, tuvieron al menos un error o carácter faltante.

Todos los conjuntos de test fueron procesados en un ordenador de escritorio cuyas características son similares a las explicadas en el apartado 6.6.1. Se han realizado tres experimentos con un número creciente de modelos. El objetivo es observar la mejora del rendimiento del método propuesto conforme va creciendo la multimodalidad al ir añadiendo nuevas evidencias. Los experimentos realizados han sido: a) considerar tan solo el modelo IM y el modelo CM (EXP1), b) añadir el modelo de error del usuario EM_{IM}, este modelo de error irá asociado al modelo IM y nos va a permitir ofrecer la tolerancia a errores por parte del sistema, pues posibilita la recuperación de éste ante errores introducidos durante el proceso de interacción (EXP2) y c) añadir además el modelo HM y EM_{HM} que se puede observar en la figura 6.2 (EXP3). El modelo EM_{IM} de EXP2 está asociado generalmente a las dimensiones

y forma del dispositivo de entrada y a errores comunes que los usuarios suelen producir al introducir datos con dichos dispositivos. Concretamente, se ofrece un modelo de confusión que considera la distribución de las teclas en el techado y donde para cada letra pulsada ofrece alternativas ponderas a ésta, dicha ponderación se realizará a través de una probabilidad que está asociada con la distancia euclídea existente entre el centroide de la letra pulsada y cada uno de los centroides del resto de las posibles letras, de manera que cuanto más cercana se encuentre la alternativa a la letra pulsada mayor será la probabilidad de confusión. Por otra parte el modelo EM_{HM} de EXP3 permite reconocer tanto municipios localizados a una determinada distancia, x km (medidos como distancia euclídea), de la longitud y latitud a la que pertenece un determinado conjunto de test, como también, municipios externos a dicho conjunto de test (EXP3).

En EXP1 se asume un escenario donde para predecir el destino se use únicamente la entrada del usuario y el modelo de restricciones. Este experimento puede considerarse como el caso base y establece el punto inicial a ir mejorando progresivamente mediante la inclusión de nuevas informaciones en el modelo. En EXP2 se incorpora el conocimiento acerca del dispositivo de entrada. Concretamente se tiene en cuenta la distribución de las teclas en el teclado y la distancia entre las mismas, lo que va a permitir la recuperación ante errores producidos por el usuario durante la introducción del destino deseado. Finalmente, en EXP3 aparece una aproximación multimodal, en la que se introducen de manera estocástica y en forma de modelo de hipótesis, las poblaciones más cercanas a una coordenada dada, ponderadas además según su población.

Con el fin de valorar el esfuerzo requerido por parte del usuario a la hora de obtener el destino deseado en cada uno de los tres modelos anteriormente planteados, se ha calculado el KSR, CSR, ASR y WKSR para todos los experimentos, tal y como se ha explicado en el apartado 6.6.1. Se ha calculado también los resultados relativos al coste temporal.

Tabla 6.4: Esfuerzo del usuario (KSR, CSR, ASR y WKSR), tasa de error y tiempo de respuesta obtenidos en cada uno de los experimentos propuestos.

	KSR	**CSR**	**ASR**	**WKSR**	**Error(%)**	**Tiempo (ms)**
EXP1	0.58	0.45	0.13	0.34	15.56	*0.05*
EXP2	0.54	0.40	0.14	0.31	4.25	0.38
EXP3	*0.25*	*0.20*	*0.05*	*0.15*	*1.17*	2.0

La tabla 6.4 muestra los resultados obtenidos en cada uno de los experimentos propuestos. Se presenta el KSR, CSR, ASR y WKSR, junto con las tasas de error y el tiempo de respuesta (expresado como el tiempo de retardo medio entre cada interacción de usuario y la obtención de una nueva propuesta). Se contabiliza un error cuando el destino propuesto no es el buscado una vez que el usuario ha introducido completamente todos los caracteres (lo cual ocurre cuando la entrada introducida por el usuario contiene errores y el sistema no es capaz de recuperarse de ellos).

Cabe destacar que aunque el 23.8 % de las entradas utilizadas en el conjunto de test contiene al menos un carácter erróneo o faltante, sólo el 15.56 % de éstos se corrigen de manera errónea en EXP1, donde no se ha aplicado ningún tipo de modelo de error. La razón de esto

es que se encuentra la cadena correcta en el modelo de restricciones antes de alcanzar al carácter incorrecto o faltante en la cadena de entrada. El proceso de entrada de datos acaba en este punto en una interfaz de usuario real. Por lo tanto, aunque no tengamos modelo de error, existe una fracción de las cadenas que serán corregidas por el propio sistema.

Si los resultados obtenidos en EXP1 representan nuestro caso base, se puede observar una mejora considerable en el rendimiento del sistema, tanto desde el punto de vista de la reducción del error como en el ahorro de pulsaciones por parte del usuario, conforme se vayan incorporando nuevos modelos que codifiquen otros tipo de fuentes de conocimiento. Se observa también que la incorporación de un modelo de error asociado a la interacción del usuario, conforme se presenta en EXP2, decrementa drásticamente la tasa de error. Cuando, conforme se ha hecho en EXP3, se construye un modelo de hipótesis a partir de una coordenada geográfica establecida previamente, la cual puede ser introducida manualmente o tomada a partir de la posición actual, entonces aparece una reducción muy significativa tanto del esfuerzo del usuario como de la tasa de error alcanzada. Esta mejora se debe, a la inclusión de nuevas evidencias sobre destinos más probables según su distancia a un origen, o su elevado número de habitantes, tal y como se comentó al comienzo del apartado, lo que incrementa su probabilidad a priori de elección frente a otros destinos.

En relación al tiempo de respuesta, se observa que en todos los casos el sistema es capaz de sugerir nuevos destinos mucho más rápido de lo que un usuario entrenado tardaría en teclear cada símbolo individual. Esto significa que la nueva sugerencia aparecerá inmediatamente, conforme el usuario teclee un nuevo símbolo. En EXP2 y EXP3 aparece un incremento del tiempo de respuesta debido a la incorporación de nuevos modelos que incrementan el número de caminos a explorar. Sin embargo, el tiempo de interacción se encuentra todavía en el rango de unos pocos milisegundos.

El radio utilizado, a la hora de construir el modelo HM en EXP3, es un factor influyente que puede afectar al rendimiento del sistema. Así pues, un radio demasiado bajo puede conducir a un modelo HM que no contenga el destino deseado, lo cual produce un incremento del KSR. Por otra parte, radios demasiado grandes producen también un incremento del KSR puesto que habrá un mayor número de destinos no deseados propuestos por el sistema como los más probables.

La figura 6.5 muestra los valores KSR, CSR y ASR en función del radio utilizado. Los destinos incluidos en cada uno de los conjuntos de test fueron construidos con un radio de 30 km, a partir de una coordenada previa elegida aleatoriamente para cada uno de los conjuntos. Esto justifica un valor mínimo de KSR en aproximadamente este nivel. Conforme más pequeño sea el radio, más baja será la probabilidad de que el destino deseado esté dentro del modelo HM y, por lo tanto, más alto será el valor del KSR. Cuando el radio vale 0, estamos en el caso en el que no se usa un modelo HM, lo que es equivalente al experimento EXP2. Por otra parte, tal y como se ha mencionado anteriormente, un radio más grande también produce un incremento del KSR, aunque en este caso el incremento de pendiente en la pérdida de rendimiento es menos severa. El radio puede ser un parámetro de usuario, puesto que el modelo HM se construye de manera muy sencilla bajo demanda cuando la interacción lo requiere.

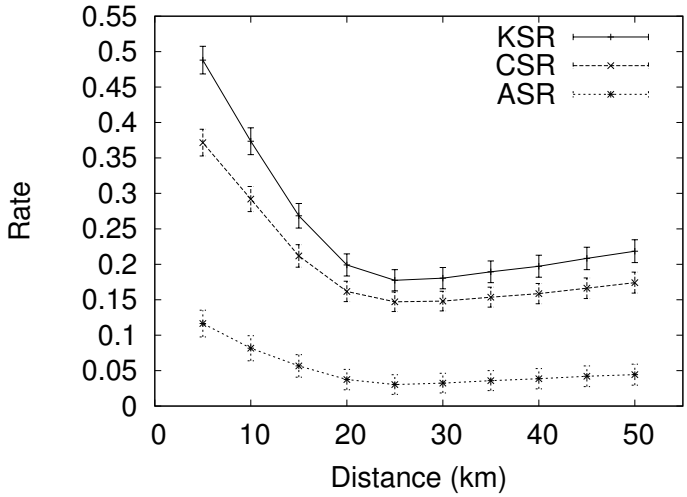

Figura 6.5: KSR, CSR y ASR frente al radio utilizado en la construcción del modelo HM junto a sus intervalos de confianza en un 95 %.

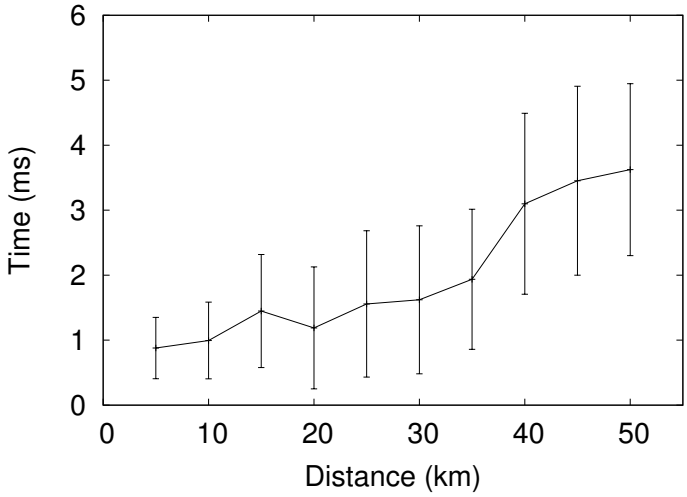

Figura 6.6: Tiempo medio de interacción respecto al radio utilizado en la construcción del modelo HM, junto a su intervalo de confianza al 95 %.

La figura 6.6 muestra la relación entre el radio utilizado para construir el modelo HM y el tiempo de respuesta alcanzado en EXP3. Como era de esperar, el tiempo de respuesta se incrementa conforme el radio va haciéndose más grande. En cualquier caso, dicho tiempo

permanece, para todos los radios probados, muy por debajo del periodo típico entre pulsaciones por parte de un usuario.

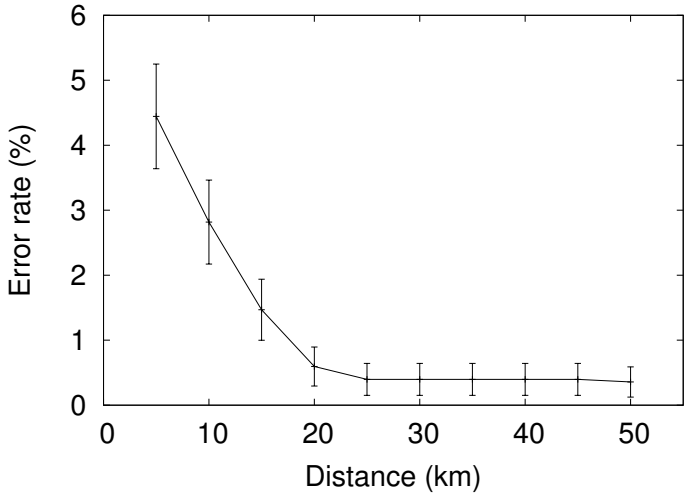

Figura 6.7: Tasa de error frente al radio utilizado en la construcción de HM, junto a su intervalo de confianza al 95 %.

Finalmente, la figura 6.7 muestra la tasa de error en el experimento EXP3 en función del radio elegido. Se puede observar que dicho error tiende al alcanzado en el experimento EXP2 conforme se decrementa el radio. Además se observa un decremento muy pronunciado del error, respecto al incremento del radio hasta llegar al valor con el que se ha creado HM (30 km), para finalmente mantener dicho error de manera estable.

6.7. Conclusiones

Se ha presentado un método genérico de postproceso interactivo multimodal de la entrada de símbolos. Para ello se ha hecho uso de transductores WFST para codificar las diferentes fuentes de información: conjunto de hipótesis de entrada, modelo de error, modelo de restricciones y modelo de interacción con el usuario. En el método que aquí se propone, no aparece un *análisis corrector estocástico* explícito de la cadena de entrada, lo que supone una clara diferencia con otro tipo de sistemas. En lugar de esto, se realiza la combinación de diferentes modelos a través de la operación de composición de transductores. La solución final consiste en la búsqueda del mejor camino, el más probable o de menor coste, dentro del transductor compuesto final.

Independientemente de la codificación de las diferentes fuentes de información de las que se disponga, la metodología propuesta permite la aplicación de una arquitectura desacoplada durante la modelización, lo que ofrece varias ventajas. En primer lugar facilita la integración

de diferentes modelos probabilísticos que representen a las fuentes de conocimiento de las que se disponga. En segundo lugar permite la aplicación de aproximaciones multimodales donde las entradas proceden de diferentes fuentes de información que necesitan ser combinadas. Finalmente ofrece una gran flexibilidad a la hora de definir, crear y combinar los distintos modelos. Además, aunque el enfoque de modelización es completamente desacoplado, se dispone de la operación de composición de autómatas que ofrece la posibilidad de unificación de todos y cada uno de los modelos planteados previamente de forma desacoplada en un único modelo, lo que supone un esquema final integrado.

Los diferentes transductores, que representan a las diferentes fuentes de información, se combinan para poder realizar una búsqueda del camino con menor coste. Como resultado de la búsqueda se ofrece la cadena más probable compatible con las restricciones (CM), las hipótesis (HM) y los diferentes modelos de error (EM_{HM}). Si como resultado se obtiene una cadena incorrecta, se puede hacer uso del modelo de interacción con el usuario (IM) junto a su modelo de error (EM_{IM}), con objeto de obtener la cadena correcta con el menor esfuerzo posible por parte del humano lo que añade interactividad y multimodalidad al sistema. En ningún momento se toman decisiones intermedias o irreversibles sobre cada uno de los modelos aislados.

Desde nuestro punto de vista, el uso independiente de modelos de error, restricciones, hipótesis de entrada y modelos de interacción con el usuario, ofrece ventajas prácticas importantes sobre otros paradigmas más acoplados tales como el Análisis Estocástico Corrector de Errores o los Modelos Ocultos de Markov.

El método propuesto es un método general que puede aplicarse directamente a un gran número de tareas diferentes. Para poder mostrar resultados con el método, se han llevado a cabo dos experimentos de diferente naturaleza. En ambos casos se han alcanzado mejoras significativas cuando se utiliza el sistema propuesto. En el caso de la tarea de postproceso OCR, la aproximación propuesta aprovecha toda la información ofrecida por el clasificador OCR, como las probabilidades *a posteriori* de las clases o la matriz de confusión de caracteres. De manera similar, en la tarea de introducción de datos GPS, se alcanza una mejora significativa cuando se añade el modelo de error asociado con la interacción del usuario o el modelo de hipótesis codificando los destinos más probables. También cabe destacar que el coste computacional es muy bajo desde el punto de vista de la interacción humana, lo que permite una interacción muy fluida con el sistema. Además, este método es aplicable fácilmente a otras disciplinas, siempre y cuando las distintas evidencias o fuentes de información disponibles puedan codificarse mediante un autómata de estados finitos.

CAPÍTULO 7

CONCLUSIONES

Índice del capítulo

En esta tesis se ha presentado un método flexible y eficiente de postproceso de cadenas que además admite la posibilidad de la interacción con el usuario. La metodología aplicada hace uso de los WFST y de la operación de composición de autómatas, gracias a la cual todas y cada una de las informaciones de las que disponemos en un momento dado y que permiten realizar el postproceso de una cadena para fines diversos, pueden modelarse de manera independiente. Ello facilita enormemente el modelado, pues nos permite centrarnos únicamente en el tipo de información a ser modelada en cada instante sin tener que considerar cualquier otro tipo de interacción con el resto de informaciones. La operación de composición de autómatas nos permite combinar todos y cada uno de los modelos independientes, en uno y más complejo que lo englobará todo como una única entidad. Así pues, la metodología que aquí presentamos tiene las mismas ventajas de simplicidad que un esquema desacoplado tiene a la hora de modelar la información junto a las ventajas de completitud que supone un esquema integrado. Para la evaluación de la metodología se han planteado dos problemas diferentes. Un primer problema inicial y más completo relativo a la mejora de la interacción humana multimodal desde la perspectiva de la reducción del esfuerzo en la digitalización de documentos y posterior validación, y un segundo problema relativo también a la mejora de la interacción humana multimodal, en este caso orientado a los dispositivos móviles.

En el primero de los problemas se hace uso de transductores WFST para un caso real de digitalización industrial. En este tipo de procesos se pueden distinguir dos fases bien diferenciadas. En una primera fase se digitaliza un campo de un formulario y mediante el uso de un motor de OCR, se obtiene una cadena reconocida que puede contener errores a nivel de carácter. En una segunda fase se postprocesa la cadena reconocida, introduciéndose una serie de restricciones e información sobre los errores tipo a nivel de carácter con el objetivo de mejorar dicha cadena inicial y reducir así la tasa de error. El proceso de mejora de la cadena ofrece como salida una nueva cadena junto a un coste de transformación. En el capítulo 4 se ofrece una información más detallada del método y las conclusiones sobre esta etapa en la sección 7.1.1.

Durante todo proceso de digitalización industrial se busca en todo momento la optimización de la productividad mediante un incremento en la calidad, entendida ésta como la digitalización de todas las cadenas con el menor número posible de errores, y una reducción del esfuerzo humano requerido. Para ello, a partir del coste de transformación asociado al proceso (relacionado directamente con el esfuerzo de transformación e inversamente con la fiabilidad de la corrección), se puede establecer un valor umbral por debajo del cual las cadenas serán aceptadas de manera automática como válidas, con la reducción del esfuerzo humano final que ello supone respecto a un proceso manual. En este tipo de procesos es deseable tener también la capacidad de negociar una tasa de error asumible (asociado a una cota sobre la calidad final) para un conjunto de cadenas digitalizadas que pertenecen a un campo del documento y por lo tanto a un lenguaje determinado. Es por ello que, en el marco de esta tesis, se ha investigado en la creación de un modelo que sea capaz de ofrecer el umbral de aceptación/rechazo tomando como entrada este error asumible. Para conseguir este objetivo se propone un modelo que representa la relación *error versus coste de transformación* y se experimenta su aplicación en dos escenarios diferentes, un primer escenario relacionado con modelos de lenguaje de los que se dispone de muestras reales etiquetadas, y un segundo escenario con modelos de lenguaje de los que no se dispone de dichas muestras. A partir de

este modelo que permite conocer la relación entre el error y el coste de transformación se establece un método para la estimación de un umbral de rechazo adaptativo, véase capítulo 5 para una información más detallada y la sección 7.1.2 para las conclusiones finales.

Una vez establecido el umbral de rechazo ya se está en disposición de aplicar dicho umbral como valor frontera para aceptar, con ciertas garantías de calidad, una serie de cadenas digitalizadas, sin requerir para ello de la supervisión humana. Esta aceptación automática de cadenas produce una reducción directa sobre el esfuerzo humano, pues se trata de cadenas directamente reconocidas por un sistema automático que no serán supervisadas por un humano. Además, en algunos sistemas de digitalización se pretende alcanzar un nivel de calidad global para todas las cadenas digitalizadas y no sólo para aquellas automáticamente aceptadas como válidas. Esto puede requerir de un proceso final de supervisión humana para todas aquellas cadenas que no superaron el control de calidad previo. En esta última fase se va a buscar también la reducción del esfuerzo humano, pero en esta ocasión se realizará a través de la incorporación en el sistema propuesto de la información proveniente de la interacción con el usuario. Lo que se busca con la introducción de dicha información es la reducción del esfuerzo humano a través de la reducción del número medio de teclas que el usuario ha de pulsar para obtener la cadena válida final. Se ha aprovechado esta última etapa de uso de los WFST en sistemas interactivos para mostrar el enorme potencial que esta metodología tiene a la hora de fusionar las evidencias, de naturaleza diversa, de las que se dispone en un momento dado. Introduciendo para ello además de la interacción, la multimodalidad en un problema relacionado con la interacción multimodal del usuario en dispositivos móviles GPS. Para una información más detallada sobre el método consultar el capítulo 6 y la sección 7.2 para las conclusiones finales.

De manera general y a modo de resumen, se puede decir que el uso de WFST presentado en el marco de esta tesis, muestra una clara ventaja respecto a otro tipo de técnicas por la simplicidad de modelización de cada tipo de información, pues cada una de ellas se define de manera independiente, conforme se haría en un sistema desacoplado, sin tener en cuenta el resto de evidencias durante dicha modelización. Como además los transductores tienen una estructura algebraica bien definida, véase capítulo 2, se pueden definir una serie de operaciones sobre éstos que incrementan el potencial de su representatividad. De entre las diferentes operaciones que se pueden aplicar en los transductores cabe destacar, por su uso continuo en el marco del trabajo que aquí presentamos, la operación de *composición*. Esta operación es la que nos va a permitir unificar toda la información creada de manera desacoplada en un esquema final completamente integrado, con las ventajas que ello conlleva. Las diferentes aplicaciones que se presentan en este trabajo muestran, de manera general, una mejora de los resultados conforme se van añadiendo las distintas evidencias haciendo uso de los WFST.

7.1. Corrección de cadenas

7.1.1. Postproceso OCR mediante el uso de WFST

Se ha propuesto un método de postproceso de cadenas procedentes de un sistema de digitalización de formularios manuscritos mediante el uso de Transductores de Estados Finitos con Pesos (WFST) y lenguajes k-testables donde pueden aparecer restricciones léxicas o lingüísticas de varios campos de manera simple o mediante la interrelación por medio de los modelos combinados. Lo que se busca con dicho postproceso es una mejora significativa en las tasas de reconocimiento, a partir de la inclusión de las diferentes evidencias relacionadas con el propio postproceso y que pueden aportar información para la obtención de la cadena correcta.

La integración de toda la información de la que se dispone en un momento dado, se realiza mediante el uso de la operación de composición de autómatas. Esta operación permite la fusión, en un esquema completamente integrado, de cada uno de los modelos previamente establecidos que codifican, de manera independiente, cada una de las fuentes de información disponibles. Este esquema integrado, a diferencia de un esquema desacoplado, contempla toda la información modelada sin ningún tipo de pérdida. La solución final al problema de corrección planteado pasa por la búsqueda del camino más probable dentro de este último autómata compuesto.

Debido al coste de la operación de composición de autómatas se hace uso de una composición *lazy* que la retrasa convenientemente, realizándose ésta durante el propio proceso de búsqueda final y sólo en el momento que se necesita y para aquellas partes del autómata completo que son objetivo de la citada búsqueda. Esta manera de proceder conlleva una reducción de los costes espaciales y temporales que hace viable la solución que aquí se presenta, consiguiéndose finalmente la cadena más probable, con tiempos de respuesta adecuados.

Se ha evaluado el comportamiento de la solución propuesta, comparando dicha solución con un método estocástico de procesado de cadenas anterior (ECP), propuesto en la literatura. Además, esta evaluación se ha realizado tanto para el caso en el que se considera únicamente el carácter más probable propuesto por el OCR en cada momento, como para el caso en el que se consideran las diferentes hipótesis ponderadas por sus probabilidades a posteriori. Los resultados muestran mejoras de la solución propuesta frente a las soluciones anteriores, y nos muestra también una mejora claramente significativa respecto del resto de casos, cuando se introducen como evidencias las propias probabilidades a posteriori del clasificador, observándose además un incremento en la importancia de la parte del modelo relativa al OCR, respecto al resto de partes, cuando se incluyen dichas probabilidades como evidencia y se optimizan dichos parámetros. Por otra parte, se ha realizado también la comparación del postproceso de las cadenas mediante el uso de campos combinados frente al uso de campos simples. Se observa una clara mejora en las tasas de reconocimiento en todos los modelos utilizados para esta comparación cuando se usan campos combinados.

Además, la flexibilidad ofrecida por los WFST, que permite modelar cada tipo de información de manera independiente, como se haría en un sistema completamente desacoplado,

sin renunciar por ello a las ventajas de completitud de un sistema integrado, hace de esta metodología un método sencillo y potente. Prueba de ello son los resultados obtenidos, anteriormente mencionados, que han mostrado una reducción significativa en las tasas de error respecto a otros tipos de metodologías sin que por ello se viesen afectados los tiempos de corrección finales.

7.1.2. Estimación del umbral de rechazo para el postproceso OCR usando Modelos de Lenguaje

Tras el proceso de corrección automática de las cadenas de un lote, es conveniente disponer de un proceso destinado a discernir las cadenas correctamente transcritas de aquellas que no lo son. Para ello disponemos, como fuente de información, del coste de transformación de la cadena de entrada en la cadena de salida. La magnitud de este coste estará directamente relacionada con el número y relevancia de las transformaciones aplicadas a la cadena origen para obtener la de destino. Así pues, a mayor coste, mayor probabilidad de obtener una transcripción errónea. Este comportamiento queda reflejado en determinadas gráficas que aparecen en el capítulo 5, donde se observa que es posible relacionar el coste de transformación con una probabilidad de error para un lenguaje dado. A partir de esta relación, y a partir de los costes de las cadenas de una muestra (o lote), se ha propuesto un método para calcular un umbral de rechazo adaptativo que garantice un error asumible por el usuario para el lote de cadenas en cuestión. El objetivo último consiste en que el usuario pueda decidir la tasa de error final para una muestra completa de cadenas y que el sistema estime el umbral de rechazo automáticamente.

La relación existente entre un umbral de rechazo y una tasa de error puede establecerse mediante el uso de herramientas analíticas tradicionales como la búsqueda del equilibrio entre el error y el rechazo, la curva *ROC* característica, o las curvas de *Precision-Recall* y sus diferentes variantes. Al utilizar este tipo de herramientas, el control de la tasa de error de una nueva muestra, conlleva la asunción de que ésta tiene una distribución de los costes de transformación similar a la utilizada durante el entrenamiento. Sin embargo, en la tarea que aquí describimos, esta asunción puede llegar a ser inaceptable, debido a que los errores OCR pueden variar de manera amplia para diferentes muestras, dependiendo de diversos factores, y consecuentemente, la distribución de los costes de transformación también variarán. Como consecuencia de esto, la aplicación de un único umbral (umbral fijo), ofrecido por tales herramientas, a muestras de diferente naturaleza, no ofrecerán suficientes garantías para la obtención de una tasa de error prefijada.

En esta tesis, se toma como hipótesis de partida el hecho de que es posible establecer una relación entre la probabilidad de error de las transcripciones y los costes de transformación producidos tras la aplicación de un modelo de lenguaje (Curva de Error vs coste (EC)), y que esta relación puede utilizarse a la hora de predecir la tasa de error de cualquier conjunto de cadenas de un lenguaje, dada la distribución de sus costes. Es por ello que se propone una técnica de umbralización adaptativa, donde la curva EC de un modelo de lenguaje se utiliza para calcular la curva acumulada del error vs coste (CEC) de una muestra cualquiera de dicho

lenguaje a partir de la cual es posible obtener de forma directa el umbral de rechazo apropiado, que nos lleva a la tasa de error prefijada por el usuario, sobre la muestra en cuestión. Lo que conllevará una garantía de calidad de las cadenas aceptadas con el esfuerzo justo a aplicar durante la corrección manual de las cadenas rechazadas, si ello es requerido.

En la presente tesis se han presentado resultados experimentales del método anterior aplicado a dos escenarios diferentes para la estimación automática de un umbral sobre el índice de fiabilidad (o coste de transformación) de una hipótesis OCR postprocesada haciendo uso de un modelo de lenguaje dado. El primer escenario hace uso de la relación existente entre la distribución de la tasa de error y los costes de transformación (curva EC) obtenidos a partir del modelo de lenguaje y un conjunto de datos procedentes de una muestra real supervisada. El segundo, sin embargo, realiza la estimación de la curva EC a partir de una muestra sintética obtenida mediante la distorsión previa de las propias cadenas del modelo de lenguaje. Esta última aproximación será de utilidad en aquellos casos en los que se quiera aplicar a un modelo de lenguaje nuevo, o una modificación sustancial de uno ya existente, para los que no se dispone de hipótesis OCR reales todavía.

En ambos escenarios, el estudio se ha realizado sobre tres estilos de escritura de calidad diferente (buena, media y mala) y sobre cuatro modelos de lenguaje de naturaleza distinta, según el número de cadenas que los conforman, la frecuencia de repetición o no de estas cadenas, la distribución de costes de transformación de sus cadenas o el error de las muestras al 0 % de rechazo. En el primer escenario se ha realizado una comparación con un método clásico de umbralización fija tanto para una muestra de test representativa del conjunto de entrenamiento, *Total Test*, como para muestras de test no representativas, *Easy Test* y *Hard Test*, donde claramente se observaba una distribución diferente de los costes de transformación. Para los tres conjuntos de test y para los cuatro modelos de lenguaje probados, la metodología de umbralización adaptativa presenta una desviación de la estimación del error muy baja, por debajo del 0.1 % respecto al error real y con intervalos de confianza muy estrechos, incluso para tasas de error objetivo altas. Para el caso de umbralización fija y el conjunto de test representativo, los resultados obtenidos relativos a la desviación de la estimación del error respecto al error real son similares al método de umbralización adaptativa para los cuatro modelos de lenguaje probados. Sin embargo, se observa un empeoramiento significativo en la desviación de la estimación del error respecto del error real cuando se usan los corpus de test no representativo y la técnica de umbralización fija, empeorando la precisión en dicha desviación de la estimación del error conforme se incrementa la tasa de error objetivo final.

En el segundo escenario se ha evaluado el sistema de umbralización adaptativa para los cuatro lenguajes anteriormente expuestos. En este caso no se disponía de muestras de test reales supervisadas, con las que estimar la curva EC, que debieron de ser generadas mediante simulación, partiendo de las propias cadenas del lenguaje mediante un método novedoso aquí propuesto. La ventaja de este segundo escenario consiste en no requerir de un conjunto de datos etiquetados previamente, lo que es particularmente beneficioso en circunstancias donde no se dispone de tales muestras etiquetadas. En este caso, la construcción de una muestra sintética etiquetada, comporta la asunción de una tasa de error intrínseca que puede ser muy diferente de una muestra de test particular lo que imposibilita el uso de una técnica de

umbralización fija para la estimación de la tasa de error. Los resultados muestran un comportamiento potencialmente útil en tareas que conlleven requerimientos de tasas de error bajos (como por ejemplo el procesamiento de formularios), y un buen rendimiento en términos de desviación de error relativo en tareas con requerimientos de tasas de error menos estrictas (como por ejemplo el OCR para la minería de datos).

La estimación del umbral de rechazo adaptativo para lotes es útil en algunas aplicaciones prácticas, como por ejemplo, en el flujo de trabajo de una industria relacionada con la entrada de datos, donde un lote de formularios u otro tipo de documentos deben de ser procesados para un cliente, y existe una tasa máxima de error aceptable que previamente ha sido negociada (control de calidad), conservando para ello la cantidad de cadenas rechazadas tan baja como sea posible (optimización de la productividad). La metodología propuesta puede ser aplicada a cualquier conjunto de observaciones que tengan un índice de confianza consistente.

7.2. Interfaces modales e interactivas

7.2.1. Mejoras en las interfaces multimodales interactivas persona-máquina mediante el uso de WFST

La obtención del umbral del coste de transformación nos permite obtener automáticamente las cadenas válidas dada la restricción del error asumido por el usuario. En algunos tipos de aplicaciones se hace una validación humana posterior de todas aquellas cadenas que no superaron el umbral, con el objetivo de corregirlas manualmente. En esta fase de validación humana toda ayuda automática que permita incrementar la productividad y la calidad será bienvenida. Es por ello que se ha presentado un método genérico de postproceso interactivo multimodal de entrada de símbolos. Para ello, se ha propuesto el uso de transductores WFST para codificar las evidencias disponibles en un momento dado. Entre estas evidencias destacan a modo de ejemplo: el conjunto de hipótesis de entrada, modelo de error, modelo de restricciones y modelo de interacción con el usuario.

Independientemente de la codificación de las diferentes fuentes de información, la metodología propuesta para la creación de una interfaz interactiva nos va a permitir el uso de una arquitectura desacoplada. Esto ofrece varias ventajas como son la facilidad de integración de diferentes modelos probabilísticos que representen a las fuentes de conocimiento de las que se disponga, además de permitir la aplicación de aproximaciones multimodales donde entradas procedentes de diferentes fuentes se fusionan posteriormente en una única con gran flexibilidad y sencillez.

En el método propuesto, los diferentes transductores, que modelan las evidencias, y que han sido creados de manera independiente, son fusionados en un único autómata transductor sobre el que se realiza la búsqueda del camino con coste más bajo. Esto ofrece como resultado la cadena más probable compatible con las restricciones (CM), las hipótesis (HM) y los diferentes modelos de error (EM_i) que pudiesen existir y que estarán asociados con el tipo de error que exista en la información disponible. Así pues, estos modelos de error pueden estar

asociados a diferentes fuentes de información, entre las que cabe destacar: las confusiones entre símbolos procedentes de un clasificador, la proximidad de las teclas en un determinado dispositivo de entrada, alteraciones en el orden de los símbolos durante la introducción de la propia cadena por parte del usuario, los distintos valores que tienen asociados una misma tecla (como ocurre en dispositivos con teclado reducido), etc.

Para validar el método propuesto se han llevado a cabo dos experimentos de diferente naturaleza, el primero relacionado con la corrección interactiva de literales y el segundo relacionado con la introducción interactiva multimodal de destinos en un navegador GPS. En ambos casos se han alcanzado mejoras significativas cuando se ha utilizado el sistema propuesto. En la tarea de postproceso OCR, la aproximación propuesta llega a considerar la información ofrecida por el clasificador OCR, como las *probabilidades a posteriori* de las clases o la matriz de confusión. De manera similar, en la tarea de introducción de datos GPS, se alcanza una mejora significativa cuando se añade el modelo de error asociado con la interacción del usuario, o incluso el modelo de hipótesis que codifica los destinos más probables. También cabe destacar que el coste computacional es lo suficientemente bajo como para permitir una interacción fluida con el sistema. Además, es un método general que puede ser aplicado fácilmente a otras disciplinas si las informaciones que éstas dispongan son modelizables de manera estructural a través de cadenas de símbolos.

Los experimentos realizados muestran que la inclusión de la información introducida en el modelo por parte del humano supone un claro incremento de la productividad, pues se va a producir, en menor o mayor medida, una reducción media del número de teclas a pulsar hasta la obtención de la cadena correcta. Este hecho queda patente en los resultados del estudio llevado a cabo al respecto en la tarea de postproceso OCR, donde se observa que considerar únicamente la propia interacción del usuario ya produce un claro decremento en los porcentajes de teclas a pulsar hasta obtener la cadena final. Por otro lado se observa que esa tasa de pulsaciones se incrementa conforme se incrementa el tamaño del lenguaje final, pues su complejidad aumenta, pero aún así en los lenguajes con mayor número de cadenas estudiados aparecen reducciones en el número de pulsaciones por encima del 40 %. Si además se opta por una aproximación multimodal, en la que se añade como fuente de información la propia clasificación del motor OCR, entonces la eficacia del sistema mejora de manera significativa. En el estudio se muestran tasas de reducción del esfuerzo necesario de hasta el 50 % para estos casos. Por otra parte, a partir de los experimentos realizados con el GPS se observa una clara reducción del error final, cuando se implementa la posibilidad de la tolerancia a fallos en la entrada. También se observa una clara reducción del esfuerzo necesario, a la hora de obtener la cadena deseada, cuando se introduce información referente a las poblaciones cercanas a una dada. Todo ello con tiempos de respuesta entre interacciones del orden de milisegundos que son tiempos de respuesta muy por debajo de los tiempos de interacción humana. Estos tiempos entre interacciones se han calculado con un ordenador de escritorio con procesador intel Xeon 2.5 GHz con 2GB de memoria.

Desde nuestro punto de vista, la introducción de las distintas evidencias mediante el uso independiente de autómatas WFST sobre los que se modela entre otros: el error, las restricciones, las hipótesis de entrada y los modelos de interacción con el usuario. Nos ofrece ventajas

prácticas importantes sobre otros paradigmas más acoplados tales como el Análisis Estocástico Corrector de Errores o los Modelos Ocultos de Markov, ya que durante el proceso, en ningún momento se toman decisiones intermedias o irreversibles, sobre los modelos aislados, debido a que estos son creados de manera independiente sin que se haya de considerar ningún tipo de interacción con otro tipo de informaciones anteriores o posteriores, y en cualquier momento se pueden modificar de manera muy sencilla sin que esto suponga cambios para el resto de modelos.

7.3. Publicaciones

Composition of Constraint, Hypothesis and Error Models to improve interaction in Human-Machine Interfaces.

- **AUTORES (p.o. de firma):** J.Ramon Navarro-Cerdan; Rafael Llobet; Joaquim Arlandis; Juan-Carlos Perez-Cortes;
  TÍTULO: Composition of Constraint, Hypothesis and Error Models to improve interaction in Human-Machine Interfaces.
  EDITORIAL: Elsevier
  REF. REVISTA (ISBN,ISSN,SUPV): Information Fusion (ISSN 1566-2535)
  VOLUMEN: 29
  PÁGINAS: 1 - 13
  AÑO: 2016
  JCR: Sí
  DOI: 10.1016/j.inffus.2015.09.001
  FACTOR DE IMPACTO: 3.681 en el año 2014

Batch-adaptive rejection threshold estimation with application to OCR post-processing.

- **AUTORES (p.o. de firma):** J.Ramon Navarro-Cerdan; Joaquim Arlandis; Rafael Llobet; Juan-Carlos Perez-Cortes;
  TÍTULO: Batch-adaptive rejection threshold estimation with application to OCR post-processing.
  EDITORIAL: Elsevier
  REF. REVISTA (ISBN,ISSN,SUPV): Expert Systems with Applications (ISSN 0957-4174)
  VOLUMEN: 22
  EJEMPLAR: 21
  PÁGINAS: 8111 - 8122
  AÑO: 2015
  JCR: Sí

DOI: 10.1016/j.eswa.2015.06.22
FACTOR DE IMPACTO: 2.240 en el año 2014

Improvement of Embedded Human-Machine Interfaces Combining Language, Hypothesis and Error Models.

- **AUTORES (p.o. de firma):** Juan-Carlos Perez-Cortes; Rafael Llobet; J.Ramon Navarro-Cerdan; Joaquim Arlandis;
  TÍTULO: Improvement of Embedded Human-Machine Interfaces Combining Language, Hypothesis and Error Models.
  EDITORIAL: Morvan, Franck and Tjoa, A Min and Wagner, Roland (DEXA)
  REF. REVISTA/LIBRO (ISBN,ISSN,SUPV): IEEE Computer Society (ISBN 978-1-4577-0982-1)
  PÁGINAS: 359 - 363
  AÑO: 2011
  DOI: 10.1109/DEXA.2011.40

OCR Post-processing Using Weighted Finite-State Transducers

- **AUTORES (p.o. de firma):** Rafael Llobet; J.Ramon Navarro-Cerdan; Juan-Carlos Perez-Cortes; Joaquim Arlandis
  TÍTULO: OCR Post-processing Using Weighted Finite-State Transducers
  EDITORIAL: IEEE Computer Society
  REF. REVISTA/LIBRO (ISBN,ISSN,SUPV): 20th International Conference on Pattern Recognition (2010) (ISBN 978-0-7695-4109-9)
  PÁGINAS: 2021 - 2024
  AÑO: 2010
  DOI: 10.1109/ICPR.2010.498

Rejection Threshold Estimation for an Unknown Language Model in an OCR Task

- **AUTORES (p.o. de firma):** Joaquim Arlandis; Juan-Carlos Perez-Cortes; J.Ramon Navarro-Cerdan; Rafael Llobet
  TÍTULO: Rejection Threshold Estimation for an Unknown Language Model in an OCR Task
  EDITORIAL: Springer
  REF. REVISTA/LIBRO (ISBN,ISSN,SUPV): 13th International Workshop on Structural And Syntactical Pattern Recognition (SSPR 2010) (ISBN 978-3-642-14979-5)
  PÁGINAS: 738 - 747
  AÑO: 2010
  DOI: 10.1007/978-3-642-14980-1_73

User-defined expected error rate in OCR postprocessing by means of automatic threshold estimation

- **AUTORES (p.o. de firma):** J.Ramon Navarro-Cerdan; Joaquim Arlandis; Juan-Carlos Perez-Cortes; Rafael Llobet
  TÍTULO: User-defined expected error rate in OCR postprocessing by means of automatic threshold estimation
  EDITORIAL: IEEE Computer Society Conference Publishing Services
  REF. REVISTA/LIBRO (ISBN,ISSN,SUPV): 12th International Conference on Frontiers in Handwritting Recognition (ISBN 978-0-7695-4221-8)
  PÁGINAS: 405 - 409
  AÑO: 2010
  DOI: 10.1109/ICFHR.2010.126

Efficient OCR Post-Processing Combining Language, Hypothesis and Error Models.

- **AUTORES (p.o. de firma):** Rafael Llobet; J.Ramon Navarro-Cerdan; Juan-Carlos Perez-Cortes; Joaquim Arlandis;
  TÍTULO: Efficient OCR Post-Processing Combining Language, Hypothesis and Error Models.
  EDITORIAL: (SSPR/SPR) Springer
  REF. REVISTA/LIBRO (ISBN,ISSN,SUPV): Lecture Notes in Computer Science, Springer (ISBN 978-3-642-14979-5)
  PÁGINAS: 728 - 737
  AÑO: 2010
  DOI: 10.1007/978-3-642-14980-1_72

Using Field Interdependence to Improve Correction Performance in a Transducer-based OCR Post-processing System

- **AUTORES (p.o. de firma):** Juan-Carlos Perez-Cortes; Rafael Llobet; J.Ramon Navarro-Cerdan; Joaquim Arlandis
  TÍTULO: Using Field Interdependence to Improve Correction Performance in a Transducer-based OCR Post-processing System
  EDITORIAL: IEEE Computer Society Conference Publishing Services
  REF. REVISTA/LIBRO (ISBN,ISSN,SUPV): 12th International Conference on Frontiers in Handwritting Recognition (ISBN 978-0-7695-4221-8)
  PÁGINAS: 605 - 610
  AÑO: 2010
  DOI: 10.1109/ICFHR.2010.99

7.4. Propiedad intelectual e industrial

Biblioteca de funciones itilmc2 para la corrección contextual de cadenas

- **INVENTORES/AUTORES/OBTENTORES (p.o. de firma):** José Ramón Navarro Cerdán; Rafael Llobet Azpitarte; Juan Carlos Pérez-Cortés; Joaquim Francesc Arlandis Navarro
  TÍTULO: Biblioteca de funciones itilmc2 praa la corrección contextual de cadenas.
  TIPO DE PROPIEDAD INDUSTRIAL: Propiedad intelectual (copyright)
  NÚMERO DE SOLICITUD: 09/2012/1265
  FECHA: 21/06/2013
  PATENTE ESPAÑOLA: Sí
  PATENTE UE: No
  PATENTE INTERNACIONAL no UE: No

Method for Symbolic correction in human-machine interfaces

- **INVENTORES/AUTORES/OBTENTORES (p.o. de firma):** Juan Carlos Pérez-Cortés; Rafael Llobet Azpitarte; José Ramón Navarro Cerdán; Joaquim Francesc Arlandis Navarro
  TÍTULO: Method for symbolic correction in human-machine interfaces.
  TIPO DE PROPIEDAD INDUSTRIAL: Patente de invención
  ENTIDAD TITULAR: ITI-Instituto Universitario Mixto Tecnológico de Informática
  NÚMERO DE SOLICITUD: P1108
  FECHA: 16/06/2011
  PATENTE ESPAÑOLA: No
  PATENTE UE: No
  PATENTE INTERNACIONAL no UE: Sí

7.5. Trabajo Futuro

El carácter general de las metodologías que se han mostrado en esta tesis, relativas éstas tanto al uso de los WFST en la mejora de las cadenas a partir de las fuentes de información disponibles, como a la estimación del error en función de la medida de confianza ofrecida por el propio proceso de corrección, permiten que puedan ser adaptadas a una gran variedad de aplicaciones que podrían ir desde el texto manuscrito continuo a la posible detección de enfermedades a través de cadenas de proteínas y cadenas de ADN o nuevos modelos de error. Como posibles trabajos futuros cabe destacar:

- Mejoras en las medidas de confianza y la estimación del error.

- Estudio y experimentación de la aplicabilidad del método de estimación del umbral adaptativo a otros problemas diferentes al postproceso OCR.
- Mejoras en el incremento de productividad durante el proceso de validación de traducciones automáticas.
- Sobresegmentación de las imágenes para el reconocimiento de texto continuo.
- Resolución de problemas de optimización de rutas en logística.
- Detección de enfermedades en cadenas de ADN, de proteínas y rutas metabólicas.

Mejoras en las medidas de confianza y la estimación del error.

En el capítulo relativo a la estimación del error, el trabajo futuro toma dos enfoques diferentes. Por una parte, introducir nuevos criterios de generación de errores en cadenas correctas para la mejora en la precisión de la estimación de la curva que relaciona el coste de corrección con el error en modelos nuevos. Por otra parte, el estudio de nuevas medidas de confianza que mejoren la separabilidad entre las distribuciones de cadenas correctamente corregidas de las que no lo han sido, permitiendo así reducir el error de tipo I (α) e incrementando la productividad para un error de tipo II (β) dado. Para tratar con el primero de los problemas se introducirán nuevas matrices que estén entre la matriz de confusión OCR y la matriz uniforme para incluir con ello otro tipo de problemáticas que pueden presentarse en las cadenas. Para el segundo de los problemas se podrían establecer nuevos estadísticos y nuevas medias de confianza a partir de estos, con la información relevante que pueda ser extraída al considerar no únicamente la información de la corrección más probable sino la resultante a partir de las n-mejores correcciones.

Estudio y experimentación de la aplicabilidad del método de estimación del umbral adaptativo a otros problemas diferentes al postproceso OCR.

El método de estimación del umbral adaptativo presentado en el marco de este trabajo es un método genérico que se puede aplicar a otros tipos de problemas, que no tienen por qué estar relacionados con la corrección OCR, y sobre los que se puede establecer una relación entre una medida de confianza o coste y el error finalmente obtenido. En esta línea se podría avanzar en el estudio y experimentación de la aplicabilidad de la técnica genérica aquí expuesta en cualquier otro tipo de aplicación que presente el tipo de relación anteriormente mencionada. Como por ejemplo las traducciones obtenidas de un traductor automático, las transcripciones realizadas a partir de un reconocedor automático del habla, etc.

Mejoras en el incremento de productividad durante el proceso de validación de traducciones automáticas.

Un traductor automático es capaz de ofrecer traducciones aprendidas automáticamente de una lengua origen en una lengua destino. En ocasiones algunas partes de las traducciones ofrecidas no son de la calidad adecuada y requieren de una postedición por parte del traductor humano. En esta tesis se ha mostrado también el uso de los WFST en tareas relativas al incremento de productividad en los procesos de validación de cadenas OCR. Estas técnicas son extrapolables a problemas de postedición de traducciones, pues se tiene un modelo de lenguaje, modelo CM, modelo de confusiones entre palabras, segmentos de palabras o símbolos, modelo EM, y las n-mejores traducciones para cada segmento dado, modelo HM. Todo ello se podría combinar junto a un modelo multimodal de interacción con el usuario que podría incluir tanto un teclado como el reconocimiento automático del habla, de manera individual o combinada, para obtener la mejor traducción posible para un segmento con baja confianza, lo que minimizaría el esfuerzo a realizar por parte del traductor.

Sobresegmentación de las imágenes para el reconocimiento de texto continuo.

Uno de los problemas que tiene el reconocimiento de texto manuscrito continuo consiste en delimitar el principio y el final de cada carácter, pues habitualmente dichos caracteres presentan enlaces o solapes que dificultan la delimitación exacta de sus fronteras. En la literatura es frecuente el uso acoplado de modelos de lenguaje junto a modelos ocultos de Markov (HMM) en el tratamiento de dicha problemática. En estos modelos HMM los caracteres se suelen plantear con una estructura secuencial de izquierda a derecha, siguiendo el sentido natural de la escritura occidental, con ciclos en los propios estados y es dentro de ellos donde se modelan las diferentes partes: inicio, centro y fin, de cada uno de los caracteres. En el interior de los estados se utilizan habitualmente modelos de mixturas de gaussinas n-dimensionales. La estimación de los parámetros que gobiernan tanto las transiciones entre estados o los ciclos al propio estado y las diferentes mixturas de gaussinas se realizan a través de un corpus de entrenamiento de imágenes etiquetadas que estiman dichos parámetros por máxima verosimilitud.

Un posible trabajo futuro, sería ofrecer otro tipo de técnicas en la resolución de tareas de reconocimiento de texto manuscrito aprovechando el potencial de los WFST. Para ello, por una parte se debería crear el modelo de restricciones CM que representará el modelo de lenguaje admitido por la tarea, el modelo EM que representará la confusión entre símbolos y el modelo HM que serán los símbolos detectados por un clasificador a diferentes niveles de segmentación y que representará nuestras hipótesis iniciales de partida.

Como inicialmente en el texto manuscrito continuo no se tiene un conocimiento a priori sobre el principio y el final de un carácter se hará uso de la sobresegmentación de la imagen junto a un clasificador para ir creando un modelo HM donde se presentarán las diferentes hipótesis de partida. Esta sobresegmentación se realizará inicialmente a varios niveles, de

mayor a menor nivel de resolución (*k* píxeles, *k+1* píxeles, *k+2* píxeles, etc.). El clasificador ofrecerá una o varias clases ponderadas para cada uno de los segmentos establecidos y sucesivamente y de manera inteligente, para que los segmentos no se solapen, se irá creando la red de estados y enlaces entre éstos que representará nuestro modelo HM completo. Este modelo HM se compondrá con nuestro modelo de confusiones EM y nuestro modelo de restricciones CM. La búsqueda del camino de máxima probabilidad ofrecerá la cadena final junto a la segmentación realizada para poder llegar a ella. Este paso nos ofrecerá un etiquetado directo de los símbolos a partir de los cuales se podrá volver a estimar el modelo clasificador y la matriz de confusión, tantas veces como sea necesario, lo que mejorará sucesivamente las estimaciones posteriores tanto del modelo HM como del modelo EM y en consecuencia nuestra clasificación final.

Resolución de problemas de optimización de rutas en logística.

Los problemas de logística se plantean como problemas en los que se quiere llegar desde un punto A a un punto B con una función objetivo determinada, menor consumo, menor tiempo, etc. y con una serie de restricciones que pueden ser estáticas cuando se conocen al principio, o dinámicas cuando éstas van apareciendo durante la realización de la ruta. Ejemplos de este tipo de restricciones pueden ser la entrega a la hora prefijada en un determinado punto, pasar antes por un destino que por otro por la propia disposición de la carga, atasco circulatorio en una determinada ruta, etc. Toda esta información puede ir representándose mediante una serie de autómatas WFST que pueden ir componiéndose conforme se ha mostrado en esta tesis. Al final la ruta óptima, desde el punto de vista de la función objetivo, será aquella que resulte de buscar el camino que mejore dicha función.

Detección de enfermedades en cadenas de ADN, de proteínas y rutas metabólicas.

Un campo potencial de posible uso de las técnicas mostradas en esta tesis sería la detección de anomalías en cadenas de ADN. En este caso el CM puede modelar las cadenas correctas para un determinado análisis, el EM, las posibles mutaciones, y el HM, la cadena real en un momento dado. El método expuesto que nos servía para corregir cadenas OCR podría extrapolarse en la búsqueda de las mutaciones más probables que aparecen en una cadena de entrada lo que nos ofrecería la posibilidad de detección precoz de enfermedades.

Otro campo de aplicación de este tipo de técnicas sería la búsqueda de anomalías en las rutas metabólicas. Las rutas metabólicas son mapas que representan una sucesión de reacciones químicas que transforman un elemento inicial, a partir de una serie de metabolitos intermediarios, en uno o varios productos finales directamente utilizados por la célula, bien para descomponer partes complejas en partes más sencillas, *rutas catabólicas*, o bien para componer partes complejas a partir de partes más sencillas, *rutas anabólicas*. Un ejemplo típico de ruta metabólica, concretamente *ruta catabólica*, sería la glucólisis que se encarga de oxidar la glucosa con la finalidad de obtener energía para el sustento de la propia célula. Las

rutas metabólicas se pueden modelar mediante el uso de autómatas no deterministas. La idea sería hacer uso de las técnicas aprendidas en esta tesis para modelar de la manera adecuada dichas rutas mediante el uso de WFST y hacer uso de las operaciones bien definidas en ellos para, por una parte detectar posibles fallos en una ruta determinada y, una vez detectados estos, determinar las posibles acciones a realizar para fomentar la producción de los elementos finales a través de posibles caminos alternativos o detectar las consecuencias de los fallos en determinados caminos para proceder a su arreglo en el caso de que ello sea factible.

APÉNDICE A

ANEXO

Índice del capítulo

A.1. Método de optimización multidimensional Downhill Simplex.

A.1.1. Introducción

Durante el desarrollo de la tesis ha sido necesaria la aplicación de un algoritmo de optimización de parámetros en los diferentes estudios presentados en ella. El algoritmo que ahora presentamos ha sido el utilizado a la hora de optimizar los parámetros en el capítulo 4, donde se ha utilizado para optimizar los parámetros del modelo corrector, en el capítulo 5 se ha utilizado para averiguar los parámetros óptimos R y P del modelo generador de cadenas sintéticas y en el capítulo 6 para optimizar los parámetros de los modelos interactivos propuestos.

El método *downhill simplex*, fue creado por Nelder and Mead Nelder and Mead [1965], y es un método geométrico para la optimización de funciones. Este método es capaz de funcionar en espacios de cualquier dimensión. Es un método robusto que no hace uso de derivadas para buscar los parámetros que minimizan la función pero ello es a expensas de la velocidad de convergencia debido al número de llamadas a la función a optimizar en comparación con otros algoritmos de optimización. La velocidad de convergencia es especialmente crítica si la dimensión del espacio de soluciones es grande. Este método de optimización utiliza un constructor geométrico llamado *simplex* para optimizar la función. Un *simplex* es una figura geométrica de N dimensiones, formada por $N + 1$ vértices que están conectados mediante líneas, caras de polígonos, etc. En dos dimensiones un *simplex* es un triángulo, en tres un tetraedro y así sucesivamente.

La manera de proceder de este método se basa en mover el *simplex* por una colina descendente, para ello los vértices de la figura, excepto el que posee el valor óptimo, van cambiando en cada iteración del algoritmo. Este algoritmo utiliza tres factores, α para movimiento de reflexión, β para movimiento de contracción y γ para movimiento de expansión. Dichos factores están relacionados pues con los tipos de operaciones de movimiento que se realizan sobre los vértices del *simplex*.

A.1.2. Descripción de las operaciones de movimiento

El método simplex en N dimensiones utiliza $N + 1$ puntos en el espacio de búsqueda para definir la figura. Se puede realizar una preselección de dichos puntos iniciales pero una asignación aleatoria permite investigar el espacio de soluciones mejor. Cada vértice de la figura *simplex* tiene un valor para la función a evaluar y los puntos con menor valor para dicha función serán mejores, pues estamos minimizando. En la evaluación de los puntos, el algoritmo crea pseudoderivadas de la función en la posición que ocupa dicha figura. La definición de derivada es:

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

En este caso los valores de h de cada derivación no se aproximan a cero pero en su lugar se tiene la distancia entre dos puntos. Esta información, es más que suficiente para el método a la hora de elegir que dirección produce mejores valores de la función objetivo. El objetivo de la función no es otro que el permitir mover los peores puntos del *simplex* a otros puntos en la búsqueda de la solución óptima.

Se buscan los valores de la función de cada uno de estos $N + 1$ puntos y se determina cuáles son: el punto con menor valor , P_L, el mayor, P_1, y el segundo mayor, P_2. Lo siguiente consiste en calcular el centroide, $\bar{P}$, de todos los puntos excepto P_1. Este, método tiene cuatro posibles pasos durante cada iteración: reflexión, contracción en una dimensión, contracción alrededor del vértice con menor valor y expansión. La descripción de cada uno de los cuatro pasos se muestra a continuación:

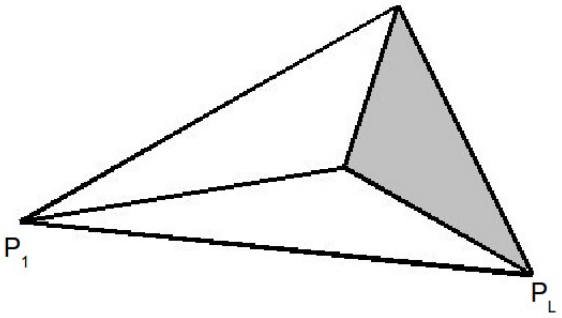

Figura A.1: *Simplex* inicial

Reflexión: Durante una reflexión, el punto de mayor valor es reflejado al lado opuesto del *simplex*. Haciendo esto el vértice de mayor valor es reemplazado por un punto que se encuentra en la dirección de máximo gradiente. Esto es una forma eficiente de que la figura *simplex* se mueva hacia valores más bajos. Un punto reflejado P_R, se consigue reflejando el punto P_1 a través del punto $\bar{P}$ haciendo uso de la ecuación:

$$P_R = (1 + \alpha)\bar{P} - \alpha P_1 \tag{A.1}$$

donde α es el factor de reflexión, en Nelder and Mead [1965], $\alpha = 1$. P_R reemplazará a P_1 si $f(P_L) < f(P_R) < f(P_1)$.

Reflexión y expansión: Para conseguir hacer las reflexiones más eficientes, se pueden establecer pesos que indican la magnitud de cada reflexión. Estos pesos pueden utilizarse a la hora de incrementar la distancia del movimiento efectivo para que sean necesarias menos iteraciones para cubrir la distancia dada. Si $f(P_R) < f(P_L)$, entonces el *simplex* crecerá a lo largo de la dirección del centroide, con la esperanza de que el punto expandido, P_E, sea mejor que P_L. La expansión se determina mediante la ecuación:

$$P_E = (1 - \gamma)\bar{P} + \gamma P_R \tag{A.2}$$

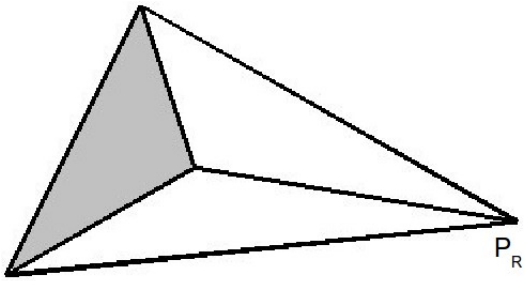

Figura A.2: *Simplex* reflejado

donde γ es el factor de expansión, en Nelder and Mead [1965], $\gamma = 2$. P_E reemplazará a P_1 si $f(P_E) < f(P_L)$.

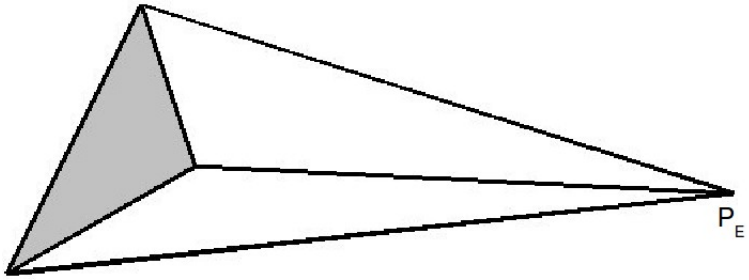

Figura A.3: *Simplex* reflejado y expandido

Contracción 1D: Existen ocasiones, cuando no se ha encontrado un mínimo, pero los mejores puntos se encuentran dentro del volumen del *simplex*. En estos casos no es deseable mover el *simplex*, pues ello produciría resultados mucho peores. En estos casos la mejor aproximación puede ser el reducir el tamaño del *simplex*. Con el método de contracción, el punto con el mayor valor es reemplazado por un punto que se encuentra más cercano a los otros vértices del *simplex*. Hacer esto no solo permite al *simplex* evaluar puntos que se encuentran normalmente en el interior del volumen, sino que además permite tener una resolución más fina en la búsqueda del óptimo. Esta operación, generalmente es útil una vez que el *simplex* está muy cerca del óptimo. Si $f(P_R) > f(P_2)$ entonces el *simplex* se contraerá a lo largo de la dirección del centroide, con la esperanza de que el punto contraído, P_C, sea mejor que P_2. Esta contracción queda determinada mediante la fórmula:

$$P_C = (1 - \beta_1)\bar{P} + \beta_1 P_0 \tag{A.3}$$

donde β_1 es el factor de contracción, en Nelder and Mead [1965], $\beta_1 = 0{,}5$ y P_0 es la selección de P_1 o P_R dependiendo de cual de los dos tiene la función con un valor más pequeño. P_C reemplaza a P_1 si $f(P_C) < f(P_0)$.

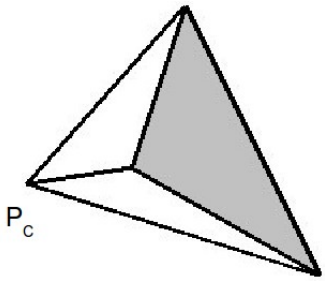

Figura A.4: *Simplex* contraído en una dimensión

Reducción: En lugar de contraer un único vértice, múltiples vértices pueden ser contraídos a la vez en una única iteración. Este es un método rápido para reducir el tamaño del *simplex*. Con esta operación el vértice con menor valor permanece fijo y el resto de puntos se mueven hacia este lugar. Si $f(P_C) > f(P_0)$ entonces la contracción anterior no será suficiente, y el *simplex* completo será contraído hacia P_L. La contracción completa queda determinada mediante la ecuación:

$$P_i = (1 - \beta_2)P_L + \beta_2 P_i \tag{A.4}$$

donde β_2 es el factor de contracción completa, en Nelder and Mead [1965], $\beta_2 = 0{,}5$ y P_i representa a todos los puntos excepto P_L.

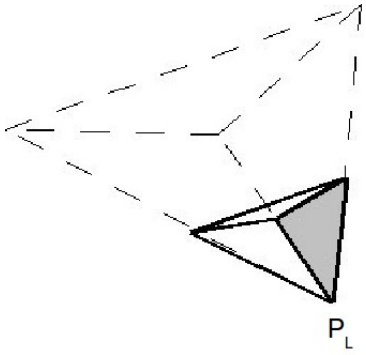

Figura A.5: *Simplex* reducido

Típicamente, cuando un punto reemplaza a P_1 la iteración actual se ha completado. A continuación se testea la condición de terminación. Si no se cumple con el nivel de tolerancia,

entonces se inicia una nueva iteración. Si el nivel de tolerancia es aceptable, entonces la optimización ya está hecha.

A.1.3. Algoritmo

1. Ordenar los puntos de acuerdo a los valores de los vértices:

$$f(x_1) \leq \cdots \leq f(x_{n-1}) \leq f(x_n)$$

$$P_L \leq \cdots \leq P_2 \leq P_1$$

2. Calcular $\bar{P}$, el centro de gravedad de todos los puntos excepto P_1.

3. **Reflexión**

   - Calcular la reflexión del punto $P_R = (1+\alpha)\bar{P} - \alpha P_1$
   - **Si** el punto reflejado es mejor que el segundo peor punto, P_2, de los $N+1$ planteados, pero no es mejor que el mejor de todos, $f(P_L) \leq f(P_R) < f(P_2)$ **entonces** obtener un nuevo *simplex* reemplazando el peor punto P_1 con el punto reflejado P_R y saltar al punto **1**.

4. **Expansión**

   - **Si** el punto reflejado es el mejor de los puntos, $P_R < P_1$, calcular el punto expandido, $P_E = (1-\gamma)\bar{P} + \gamma P_R$
     - **Si** el punto expandido es mejor que el punto reflejado, $P_E < P_R$, **entonces** obtener un nuevo *simplex* reemplazando el peor punto P_1 con el punto expandido P_E, y saltar al punto **1**.
     - **sino** obtener un nuevo *simplex* reemplazando el peor punto P_1 con el punto reflejado P_R, y saltar al punto **1**.
   - **sino** saltar al punto 5.

5. **Contracción**

   - **Si** $P_R \geq P_2$, calcular el punto contraído $P_C = (1-\beta_1)\bar{P} + \beta_1 P_0$. P_0 es la selección de P_1 o P_R dependiendo de cual de los dos tiene la función con un valor más pequeño.
     - **Si** el punto contraído es mejor que el peor punto, $P_C < P_1$ **entonces** se obtiene un nuevo *simplex* reemplazando el peor punto, P_1 por el punto contraído, P_C, y saltar el punto 1.
   - **sino** saltar al punto 6.

6. **Reducción**

- Para todos los puntos excepto el mejor de ellos, reemplazar el punto por $P_i = (1-\beta_2)P_L + \beta_2 P_i$, y saltar al punto 1. P_i representa a todos los puntos excepto P_L

La figura *simplex* inicial es muy importante y también lo son los parámetros[a] que influyen en las operaciones que se pueden realizar sobre dicha figura con el objeto de que busque el mínimo.

Para la **reflexión**, puesto que P_1 es el vértice con el mayor valor asociado de entre todos los vértices, es de esperar el encontrar un punto con un valor más pequeño haciendo uso de esta operación de reflexión y obteniendo una figura completamente simétrica sin dicho punto máximo.

Para la **expansión**, si el punto reflejado es un nuevo mínimo a lo largo de los vértices es de esperar el encontrar puntos interesantes que siguen esta dirección y que se encuentren más alejados.

Respecto a la **contracción**, si $P_R > P_2$, se espera que el mejor valor se encuentre en la parte interna de la figura *simplex*.

A.2. Modelo de Regresión.

A.2.1. Introducción

En el capítulo 5 se ha hecho uso de un modelo de regresión a la hora de obtener la función objetivo que nos relaciona los parámetros R y P del modelo generador de muestras sintéticas, con un estadístico que ofrecía información acerca de la aproximación existente entre la *curva de error versus coste de transformación*, de la muestra de test con la muestra sintética. Lo que se persigue con este modelo es buscar la relación existente entre dichos parámetros con la función objetivo. Tras obtener dicha relación se ha usado el método de optimización del anexo A.1 para obtener los valores R y P que minimizan dicho objetivo. En este apartado se presentan los conceptos de un modelo de regresión desde su formulación inicial hasta su análisis de significación, pasando por su estimación mediante el método de mínimos cuadrados.

Tal y como se comenta en Villafranca and Ramajo [2005] y en Evans and Rosenthal [2005], los Modelos de Regresión Lineal establecen relaciones entre la variabilidad de una variable aleatoria y los valores de una o más variables de las que en principio depende la primera de ellas.

Los modelos de regresión son modelos generales que resultan especialmente interesantes cuando o bien no se pueden prefijar los valores de las variables explicativas, pues no es posible controlarlas de manera ortogonal conforme se haría en un diseño de experimentos, o bien dichas variables explicativas proceden de datos históricos anteriores.

[a] Los valores típicos de α, γ, β_1, β_2 son: $\alpha = 1$, $\gamma = 2$, $\beta_1 = 0{,}5$, $\beta_2 = 0{,}5$

En un estudio de regresión se dispone de J observaciones de una variable aleatoria Y_j junto con los valores de I variables $X_{1j}, \ldots, X_{Ij}$, y se trata de estudiar las relaciones existentes entre la distribución de Y_j y los valores de las diferentes X_{ij}. A la Y se le denomina como **variable dependiente o endógena** mientras que las variables X_{ij} se las conoce como **variables independientes, exógenas o explicativas**.

Los modelos clásicos de regresión asumen que cada observación y_j es el valor observado de una variable aleatoria Y_j normal, de varianza $\sigma^2(Y_j)$ constante y desconocida y cuyo valor medio es una función de los los valores de las variables independientes.

A.2.2. Modelado matemático

Sea Y una variable dependiente a estimar. Sean $(X_1, \ldots, X_i, \ldots, X_I)$, I variables explicativas o independientes, capaces de explicar cierto grado de la variabilidad presente en y. Se plantea un modelo de regresión como una ecuación capaz de estimar el valor esperado de Y $(E(Y))$ en función de las variables explicativas, tal y como se observa en A.5.

$$E(Y \| X_1, \ldots, X_i, \ldots, X_I) = \beta_0 + \beta_1 X_1 + \cdots + \beta_i X_i + \cdots + \beta_I X_I \tag{A.5}$$

Sea x_{ij}, el valor que toma la variable X_i en un instante j. Y sea y_j el valor que el modelo estima para la variable dependiente Y a partir de los valores observados para cada una de las variables independientes en el instante j. La ecuación A.6 nos muestra dicha estimación en el instante j.

$$y_j = \beta_0 + \beta_1 x_{1j} + \cdots + \beta_I x_{Ij} + \varepsilon_j \tag{A.6}$$

donde, ε_j son valores independientes que se distribuyen como $N(0, \sigma^2)$. A partir de N observaciones de las variables independientes junto a la observación de la variable dependiente, donde $N > I$, se estimará el modelo de regresión tal y como se muestra a continuación:

$$\vec{y} = \begin{bmatrix} y_1 \\ \ldots \\ y_j \\ \ldots \\ y_N \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{i1} & \ldots & x_{I1} \\ 1 & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{1j} & \ldots & x_{ij} & \ldots & x_{Ij} \\ 1 & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{1N} & \ldots & x_{iN} & \ldots & x_{IN} \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \ldots \\ \beta_j \\ \ldots \\ \beta_N \end{bmatrix} \qquad \vec{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \ldots \\ \varepsilon_j \\ \ldots \\ \varepsilon_N \end{bmatrix}$$

donde $\vec{y}$ serían los N valores de la muestra de la variable dependiente y $\mathbf{X}$ contiene los N valores de la muestra, alineados con los N valores obtenidos para $\vec{y}$, para una variable independiente $I-$dimensional.

$$\vec{y} = \mathbf{X}\vec{b} + \vec{\varepsilon} \tag{A.7}$$

A partir de esta información el proceso de estimación realizará la estimación de los valores b_i para cada uno de los $I+1$ parámetros β_i, además de las estimación s_{b_i} de dichos parámetros (β_i), como medida del margen de incertidumbre asociado a cada estimador. También se realizará la estimación de la varianza residual σ^2 a partir de s^2_{n-1} del modelo. Para la realización de tales estimaciones, se ha de definir el residuo e_j para cada observación, como la diferencia entre el valor real observado y el valor estimado por el modelo, tal y como se muestra en la ecuación A.8.

$$e_j = y_j - (b_0 + b_1 x_{1j} + \cdots + b_I x_{Ij}) \tag{A.8}$$

Desde un punto de vista de las propiedades estadísticas, los $\vec{b} = b_0, b_1, \ldots, b_I$ óptimos, son los que conducen a un valor mínimo de la suma de los cuadrados de dichos residuos tal y como muestra la ecuación A.9.

$$\vec{b} = b_0, b_1, \ldots, b_I | \min \sum_{j=1}^{N} \left(y_j - (b_0 + b_1 x_{1j} + \cdots + b_I x_{Ij})\right)^2 \tag{A.9}$$

que matemáticamente se calculan como se muestra en la ecuación A.10.

$$\vec{b} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \vec{y} \tag{A.10}$$

La varianza residual (σ^2) del modelo se estima tal y como se muestra en la ecuación A.11 y equivale a lo que se conoce como cuadrado medio residual ($\text{CM}_{Residual}$).

$$s^2 = \frac{\sum_{j=1}^{N} e_j^2}{N-1-I} = \frac{\sum_{j=1}^{N} \left(y_j - (b_0 + b_1 x_{1j} + \cdots + b_I x_{Ij})\right)^2}{N-1-I} \tag{A.11}$$

La desviación típica de cada uno de los parámetros b_i, que es una medida del margen de incertidumbre asociado a cada estimador, se calcula tal y como se muestra en la ecuación A.12.

$$s_{b_i} = s\sqrt{c_{ii}} \tag{A.12}$$

donde c_{ii} son los elementos de la diagonal principal de la matriz $\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}$.

A.2.3. Significación estadística de una variable en el modelo

A.2.3.1. Distribución χ^2

El estudio de la distribución de la varianza muestral de una población normal nos lleva a la distribución χ^2. Por definición, una variable Y sigue una distribución χ^2 con ν grados de libertad si es el resultado de la suma de los cuadrados de ν variables independientes que se distribuyen según una distribución $N(\mu = 0, \sigma = 1)$.

Sea $X_1, \ldots, X_\nu$ variables independientes $N(\mu = 0, \sigma = 1)$ entonces:

$$Y = X_1^2 + \cdots + X_\nu^2 \sim \chi_\nu^2$$

A continuación se muestra una distribución χ^2 con $\nu = 5$ grados de libertad.

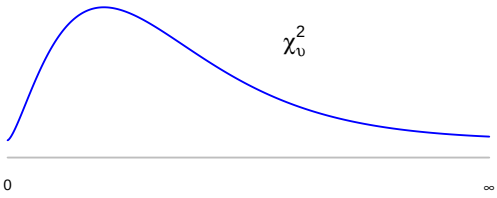

Figura A.6: Distribución χ^2 con 5 grados de libertad

En inferencia estadística la importancia de esta distribución se debe a que si S^2 es la varianza muestral de una muestra de tamaño n extraída de una población normal de varianza σ^2, entonces se demuestra que:

$$(n-1)\frac{s^2}{\sigma^2} \sim \chi_{n-1}^2 \tag{A.13}$$

el resultado del estadístico presente en la ecuación A.13 nos permite estimar la varianza poblacional (σ^2), que no es medible directamente, a partir de parámetros muestrales que si que lo son.

A.2.3.2. Distribución T-Student

En la estimación de la media poblacional en distribuciones normales la distribución t-Student juega un papel fundamental. Esta distribución fue inventada por el matemático William Sealy Gosset quien se vio obligado a firmar sus trabajos con el pseudónimo de Student.

Así pues, sea X una variable que se distribuye mediante una distribución normal tipificada, $N(\mu = 0, \sigma = 1)$, e Y una variable que se distribuye conforme a una distribución χ^2 con

υ grados de libertad y que es independiente de la anterior distribución, entonces el estadístico:

$$t = \frac{X}{\sqrt{\frac{Y}{\upsilon}}} \tag{A.14}$$

se distribuye conforme a una distribución t-Student con υ grados de libertad (t_υ). Esta distribución es simétrica y está centrada en 0, $\mu = 0$, su varianza es $\sigma^2 = \frac{\upsilon}{\upsilon - 2}$. Cabe destacar que conforme más grados de libertad tenga esta distribución la varianza se aproximará a 1 con lo que se aproximará a una distribución normal tipificada $N(\mu = 0, \sigma = 1)$ tal y como puede apreciarse en la figura A.7.

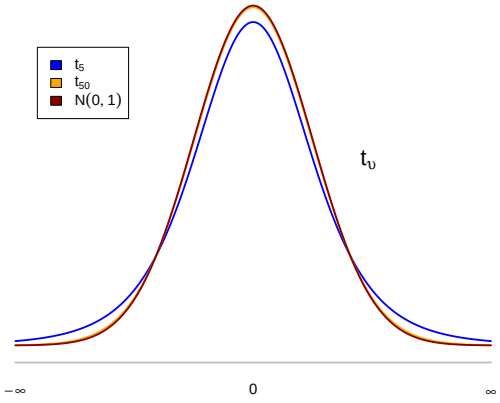

Figura A.7: Distribución t-Student con 5 y 50 grados de libertad. Se observa que conforme aumentan los grados de libertad la distribución se aproxima a una distribución Normal tipificada

A partir de este modelo teórico lo que justifica la relevancia de la distribución t de Student en lo referente a la teoría del muestreo es el estadístico:

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}} \sim t_{n-1} \tag{A.15}$$

donde, $\bar{x}$ es la media muestral, s la desviación típica muestral y n el tamaño de la muestra.

En la fórmula anterior, se observa que en el numerador existe una diferencia entre el parámetro de posición muestral $\bar{x}$ que somos capaces de medir a partir de una muestra de tamaño n y el parámetro de posición poblacional μ que representará nuestra hipótesis de partida. En el denominador se tiene la estimación de la desviación típica muestral del parámetro de posición $\bar{x}$, para una muestra de tamaño n. Este denominador nos ayuda a tipificar el resultado

final de manera que los valores obtenidos se puedan reflejar en alguna zona de la distribución t-Student concreta.

A partir de esto, la manera de proceder en inferencia estadística consiste en establecer una hipótesis inicial H_0, que será nuestra hipótesis nula. Esta hipótesis nula será una hipótesis conservadora, pues se suele utilizar para suponer lo que habitualmente esperamos. Aquí asumiremos que $H_0 : \mu = cte$ y siempre tendremos una hipótesis alternativa, H_1, que representará lo contrario de lo que representa H_0, en el caso que nos ocupa $H_1 : \mu \neq cte$. Así pues, lo que encontraremos en inferencia estadística es un duelo entre dos partes en la que una de las hipótesis resultará vencedora.

Para poder establecer la hipótesis ganadora, necesitaremos hacer uso del estadístico presente en la ecuación A.15 además de la distribución gráfica de la t-Student correspondiente al tamaño de muestra utilizada en nuestro estudio. Para ello, se ha de establecer previamente un riesgo de primera especie (α), que representará la probabilidad de decidir que la hipótesis ganadora es la H_1 cuando realmente es la H_0 lo que realmente está ocurriendo.

A partir de este riesgo α vamos a ser capaces de crear dos zonas en la t-Student correspondiente, que nos delimitan la zona de cercanía entre lo observado muestralmente ($\bar{x}$) y lo supuesto poblacionalmente (μ), y que se corresponde con la zona de aceptación de la hipótesis H_0, de la zona de lejanía y que se corresponde con la zona de aceptación de la hipótesis H_1.

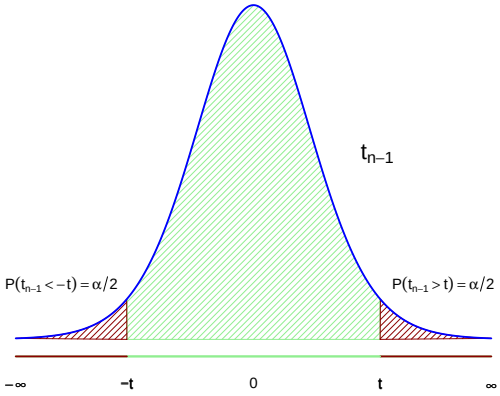

Figura A.8: Zona de aceptación y rechazo en una t-Student con $n-1$ grados de libertad. La zona central sería la zona de aceptación de la hipótesis H_0 mientras que las colas de los extremos corresponden a la zona de aceptación de la hipótesis H_1 para un riesgo de primera especie α

.

Dado un modelo de regresión como el mostrado en la ecuación A.16, y una estimación para cada uno de los parámetros β_i que lo forman, mediante la metodología expuesta en la sección A.2.2

$$E(Y) = \beta_0 + \beta_1 X_1 + \cdots + \beta_i X_i + \cdots + \beta_I X_I \tag{A.16}$$

El modelo de regresión final ha de contener los parámetros β_i que realmente resulten significativos estadísticamente. Para ello, se hará uso de la inferencia estadística estableciendo las siguientes hipótesis:

$$H_0\text{: } \beta_i = 0$$
$$H_1\text{: } \beta_i \neq 0$$

A partir de aquí, si la hipótesis H_0 resulta ganadora, entonces se asume que el parámetro β_i no es significativo estadísticamente y por lo tanto se restimará el modelo sin la variable explicativa asociada a dicho parámetro. El modelo final deberá contener únicamente aquellas variables explicativas que hayan resultados estadísticamente significativas, y la estructura resultante será la que explique la naturaleza de la variable dependiente en función de las variables explicativas presentes finalmente en el modelo.

Para poder realizar la inferencia sobre cada uno de los parámetros presentes en el modelo, se partirá de los parámetros, b_i y s_{b_i}, obtenidos a partir de las N muestras para cada una de las I variables explicativas, estimados en las ecuaciones A.10 y A.12 respectivamente. Para ello se hace uso del estadístico mostrado en la ecuación A.17

$$\frac{b_i - \beta_i}{s_{b_i}} \underset{\beta_i=0}{=} \frac{b_i}{s_{b_i}} \sim t_{N-1-I} \tag{A.17}$$

A partir del valor obtenido para el estadístico procedente de la ecuación A.17, se establecerá un riesgo de primera especie α y se verá si el valor de este estadístico cae dentro de la zona de aceptación de H_0 o de aceptación de H_1, tal y como se muestra en la figura A.9, asumiendo como cierta la hipótesis que resulte de dicho análisis.

A.2.4. Significación global del modelo

A.2.4.1. Distribución F-Fisher

Sean las variables X_1 y X_2 dos variables χ^2 independientes con ν_1 y ν_2 grados de libertad ($\chi^2_{\nu_1}$, $\chi^2_{\nu_2}$). Bajo este supuesto el estadístico:

$$f = \frac{\frac{X_1}{\nu_1}}{\frac{X_2}{\nu_2}} \tag{A.18}$$

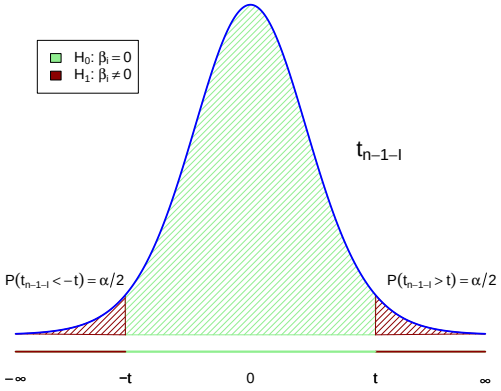

Figura A.9: Distribución t-Student para β_i y riesgo de primera especie α

sigue un distribución F con ν_1 y ν_2 grados de libertad (F_{ν_1,ν_2}). Tal y como puede apreciarse en la figura A.10 se trata de una distribución con asimetría positiva. Esta asimetría irá decreciendo conforme aumenten los grados de libertad del numerador y denominador. La media de esta distribución es ≈ 1 ($E(f) = \frac{\nu_2}{\nu_2 - 2}$).

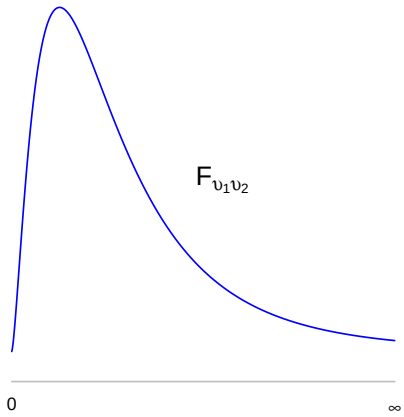

Figura A.10: Distribución F-Fisher con ν_1 y ν_2 grados de libertad.

Concretamente, la distribución F se utiliza en la práctica para comparar la variabilidad entre dos poblaciones distintas. Así pues, sea s_1^2 y s_2^2 la cuasivarianza de dos muestras de tamaño n_1 y n_2 respectivamente, extraídas éstas de dos poblaciones normales independientes con varianza poblacional σ_1^2 y σ_2^2. Bajo este supuesto el estadístico:

$$f_{calc} = \frac{\frac{s_1^2}{\sigma_1^2}}{\frac{s_2^2}{\sigma_2^2}} \sim F_{n_1-1,n_2-1} \tag{A.19}$$

se distribuye conforme a una F_{n_1-1,n_2-1}. A partir de este estadístico, y puesto que los parámetros poblaciones son parámetros desconocidos se puede hacer uso de la inferencia estadística, mediante el contraste de hipótesis, para contrastar las varianzas poblacionales de ambas poblaciones. Estableciendo para ello las siguientes hipótesis:

$$H_0\text{: } \sigma_1^2 = \sigma_2^2$$
$$H_1\text{: } \sigma_1^2 > \sigma_2^2$$

El método de inferencia establece que si la suposición H_0 es cierta entonces:

$$f_{calc} = \frac{\frac{s_1^2}{\sigma_1^2}}{\frac{s_2^2}{\sigma_2^2}} = \frac{s_1^2}{s_2^2} \sim F_{n_1-1,n_2-1} \tag{A.20}$$

y si no es cierta, y se ha tenido el cuidado de introducir en el numerador la cuasivarianza mayor y en el denominador la menor, entonces el estadístico f_{calc} obtenido será mayor que una F_{n_1-1,n_2-1}. Para dar un sentido objetivo al concepto "mayor" en inferencia estadística se asume un riesgo de primera especie $\alpha = P(H_1|H_0)$, que en el caso que nos ocupa es la probabilidad de aceptar que las varianzas son distintas cuando realmente son iguales, y se averigua cuál es el valor $f : P(F_{n_1-1,n_2-1} > f) = \alpha$. Una vez averiguado dicho valor f, este será el umbral de tal manera que si f_{calc} cae a la izquierda de f se considerará H_0 como cierta mientras que si f_{calc} cae a la derecha será H_1 la considerada como cierta (véase figura A.11).

Por otra parte, el valor umbral f depende de los grados de libertad, asociados al número de evidencias de cada una de las muestras. Cuanto mayor sea este número de muestras (n_1, n_2) se tendrá una mayor potencia, o capacidad de detectar H_1 cuando ésta realmente es cierta (véase anexo A.5). Una mayor potencia redundará en un menor valor umbral f ante el mismo riesgo α. Así pues una mayor potencia estará asociada, en este caso, a poder afirmar de manera estadística y con un determinado riesgo α que H_1 es cierto con un ratio entre cuasivarianzas menor que si el experimento es menos potente. En la figura A.12 se puede apreciar como los valores umbrales f, para dos distribuciones que se diferencian en el número de muestras del denominador n_2, es menor en la distribución con n_2 mayor.

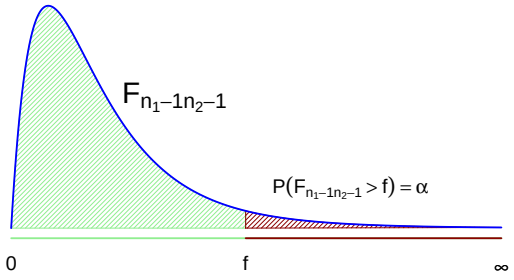

Figura A.11: Distribución F-Fisher con n_1 y n_2 grados de libertad y riesgo de primera especie α

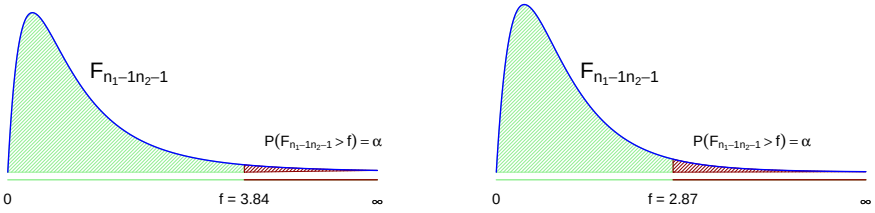

Figura A.12: La figura de la izquierda muestra una distribución F-Fisher con $n_1 = 4$ muestras y $n_2 = 8$ muestras y riesgo de primera especie α =0.05, mientras que la figura de la derecha nos muestra una distribución F-Fisher con $n_1 = 4$ muestras y $n_2 = 20$ muestras para el mismo riesgo de primera especie anterior.

A.2.4.2. Análisis de la varianza de un modelo de regresión

En un modelo de regresión, la variabilidad total se puede descomponer en dos tipos de variabilidad diferentes que son: por una parte la variabilidad explicada por el modelo y por otra la variabilidad residual que coincidirá con la explicada por el resto de variables que no han sido incluidas en éste. Bajo esta perspectiva, en un modelo de regresión tendremos dos tipos diferentes de variabilidad que podemos medir y comparar, una vez estimado éste, para averiguar si el modelo estimado es o no significativo estadísticamente en un sentido global.

Así pues, sean s_t^2, s_e^2, s_r^2, las quasivarianzas, total, explicada y residual, del modelo de regresión en cuestión, y que pueden ser calculadas a partir de los valores de la propia muestra

de entrenamiento y las estimaciones ofrecidas por el modelo para estas mismas muestras. Sean también gl_t, gl_e, gl_r, los grados de libertad totales, del modelo y residuales, respectivamente, tal que $gl_t = gl_e + gl_r$, donde: $gl_t = n - 1$, siendo n el número de muestras con el que se ha estimado el modelo, $gl_e = I$, siendo I el número de factores incluidos en el modelo y $gl_r = gl_t - gl_e = n - 1 - I$. En principio para que un modelo sea significativo la $s_e^2 > s_r^2$, pues evidentemente si un modelo estimado es significativo estadísticamente entonces la varianza explicada por éste deberá de ser mucho mayor que la variabilidad residual. Bajo este supuesto, se establecen las siguientes hipótesis:

$$H_0\text{: } \sigma_e^2 = \sigma_r^2$$
$$H_1\text{: } \sigma_e^2 > \sigma_r^2$$

donde H_0 estaría diciéndonos que el modelo establecido no explica más que lo que sería capaz de explicar el resto de factores no incluidos en dicho modelo y por lo tanto nuestro modelo no sería significativo mientras que admitir H_1 implicaría la aceptación del modelo como estadísticamente significativo.

A partir de la ecuación A.20 del apartado anterior, se puede calcular, $f_{calc} = \frac{s_e^2}{s_r^2}$, que en este caso y en el supuesto de que $H_0 : \sigma_e^2 = \sigma_r^2$ sea cierta, se distribuirá conforme a una F_{gl_e, gl_r}. A partir de aquí dado un riesgo α y operando conforme se expuso en el anterior apartado se podrá decidir desde un punto de vista objetivo si el modelo es significativo estadísticamente a nivel global o no lo es.

A.3. Diseño de experimentos para la generación de un modelo que permita la estimación de los parámetros en la generación de muestras sintéticas.

En el capítulo 5 se ha planteado un diseño de experimentos para estudiar el efecto de los factores R y P sobre la función objetivo perseguida en el modelo generador de muestras sintéticas. Tal y como se comentó en el apartado 5.7, el fin del experimento consiste en encontrar un modelo que relacione la función objetivo que se muestra en la ecuación 5.7 con los parámetros R y P. Inicialmente, se puede partir de un modelo con un grado máximo, como el que se muestra en la ecuación 5.8, y mediante las técnicas de análisis estadístico, expuestas en el anexo A.2.3 y A.2.4, ir refinando dicho modelo inicial hasta conseguir el modelo estadísticamente significativo que mejor explique a nuestra función objetivo. Si observamos el modelo se puede apreciar que las variables explicativas del modelo planteado son los parámetros R y P, bien como factores simples, lineales o cuadráticos o como interacción entre ambos. Estos parámetros son los que gobiernan el modelo generador de muestras sintéticas planteado en el apartado 5.7. Así pues, lo que se pretende al estimar este nuevo modelo, capaz de explicar parte de la variabilidad de nuestra función objetivo inicial, no es otra cosa que el obtener una expresión matemática que permita la optimización de dichos parámetros. Tras esto, mediante el algoritmo expuesto en el anexo A.1, se buscarán los valores de R y P para los que el

modelo tiene un mínimo local, tal y como se muestra en la ecuación 5.9. Esta optimización se podrá realizar de manera individual, para cada uno de los lenguajes, o de manera general, para el conjunto completo de lenguajes. Así pues, sea:

P: Probabilidad asignada a la diagonal de la matriz uniforme. El resto de masa de probabilidad, $1 - P$, será distribuida uniformemente entre el resto de elementos de la fila (o columna) que no se corresponden con el elemento que hay en la diagonal.

R: Porcentaje de cadenas distorsionadas con la matriz OCR. El porcentaje restante de cadenas, $1 - R$, será distorsionada con la matriz uniforme.

Como se desea dar la posibilidad de estudiar una posible relación no lineal entre la variable dependiente y las variables explicativas, se plantea un modelo que es capaz de modelizar un posible comportamiento de segundo orden, tal y como se muestra en la ecuación A.21.

$$J = \beta_0 + \beta_1 P + \beta_2 R + \beta_3 P \cdot R + \beta_4 P^2 + \beta_5 R^2 + \varepsilon \tag{A.21}$$

que puede ser expresado según el valor esperado, $E(\cdot)$, como:

$$\begin{aligned} E(J|R,P) = &\beta_0 + \beta_1 P + \beta_2 R + \\ &\beta_3 P \cdot R \\ &\beta_4 P^2 + \beta_5 R^2 \end{aligned} \tag{A.22}$$

donde:

β_0: Valor esperado de J en el origen, es decir cuando $P = 0$ y $R = 0$.

β_1: Incremento del valor esperado de la variable J por cada incremento unitario de P en el origen, manteniendo constantes el resto de variables.

β_2 : Incremento del valor esperado de la variable J por cada incremento unitario de R en el origen, manteniendo constantes el resto de variables.

β_3 : Incremento que se produce en el incremento del valor esperado de la variable J por incremento de P cuando se produce un incremento unitario de R.

β_4 : Incremento de la pendiente del valor esperado de J por cada incremento unitario de P, manteniendo constantes el resto de variables.

β_5 : Incremento de la pendiente del valor esperado de J por cada incremento unitario de R, manteniendo constantes el resto de variables.

Así pues, tal y como puede observarse en la ecuación A.21, el modelo cuadrático planteado requiere de la estimación de un total de cinco parámetros (β_i). Ello supone la necesidad de gastar un total de 5 grados de libertad, de los grados de libertad totales que haya en el experimento, uno por cada uno de los parámetros a estimar que tenga el modelo.

Por otra parte, y debido a la necesidad de que el experimento tenga la potencia [b] suficiente (véanse los anexos A.2 y A.5), existe la necesidad de tener un mínimo de 4 grados de libertad residuales. Un incremento en el número de grados de libertad supone un incremento no lineal en la potencia del experimento.

Otro tema relacionado con el número de niveles con los que experimentar cada uno de los parámetros estará relacionado con el orden máximo a modelizar para dicho parámetro. Así pues, para poder modelizar un orden n sobre un parámetro, será necesario plantear como mínimo $n + 1$ niveles distintos sobre dicho parámetro. Para poder estimar un efecto lineal los parámetros necesitan como mínimo dos niveles, si lo que se desea es estudiar un efecto de segundo orden entonces será necesario que los parámetros tengan como mínimo 3 niveles diferentes y así sucesivamente. Debido a que este número de niveles para los parámetros es un mínimo y adelantándonos a la posibilidad de un estudio posterior con un modelo más complejo con estructura cúbica, se plantean 4 niveles equidistantes diferentes para cada uno de los parámetros a estimar tal y como se muestra en la tabla A.1.

Tabla A.1: Valores de los niveles en el diseño de experimentos planteado

	P	**R**
-2	0,05	0,05
-1	0,35	0,35
1	0,65	0,65
2	0,95	0,95

Finalmente, se han elegido un total de 16 tratamientos ortogonales por modelo, tal y como se muestra en la tabla A.2. En el caso de realizar la estimación para cada uno de los modelos de lenguaje por separado tendríamos un total de 15 grados de libertad de los cuales 6 irán destinados a la estimación de los parámetros del modelo completo con lo que nos quedarán un total de 9 para el residuo lo que nos ofrece una potencia del experimento más que suficiente para lo que aquí se pretende. Sin embargo si nuestra intención es realizar un único modelo general para todos los modelos de lenguaje, cuatro en nuestro caso, estaríamos en la situación de un diseño de experimentos con 4 réplicas y tendríamos un total de 16 tratamientos repetidos cada uno 4 veces, uno por cada modelo de lenguaje lo que nos ofrecería un total de 63 grados de libertad de los cuales 6 estarían dedicados a la estimación del modelo completo y 57 de ellos serían residuales.

Una vez decididos los tratamientos se comienza con la experimentación para obtener el valor de la función objetivo que se muestra en la ecuación 5.7 y que será el valor de la variable dependiente para cada uno de los tratamientos.

[b] Probabilidad de detección de la significación real de un factor.

Tabla A.2: Tratamientos elegidos para el experimento que permitirá realizar la estimación del modelo A.21

P	-2	-2	-2	-2	-1	-1	-1	-1	1	1	1	1	2	2	2	2
R	-2	-1	1	2	-2	-1	1	2	-2	-1	1	2	-2	-1	1	2

Tras haber realizado todos los experimentos se pasa a la fase de estimación del modelo por el método de mínimos cuadrados, junto a su posterior validación estadística cuyo objetivo no es otro que el de averiguar que parámetros β_i son realmente significativos y cuales no lo son. Esto nos ofrecerá, conforme se comenta en el anexo A.2, la mejor subestructura, a partir de la estructura del modelo completo, que ofrezca una respuesta estadísticamente significativa, en caso de haberla, para la variable dependiente que nos ocupa.

El resultado final de esta estimación fue que el modelo que realmente resultaba significativo tenía la forma:

$$E(J|R,P) = \beta_0 + \beta_1 P + \beta_2 R + \beta_3 P \cdot R \tag{A.23}$$

cuya nueva subestructura hace cambiar sutilmente la interpretación de alguno de los parámetros presentes en el nuevo modelo:

β_0: Valor esperado de la variable J en el origen, cuando $R = 0$ y $P = 0$.

β_1: Incremento del valor esperado de la variable J por cada incremento unitario de P, manteniendo constantes el resto de variables.

β_2 : Incremento del valor esperado de la variable J por cada incremento unitario de R, manteniendo constantes el resto de variables.

β_3 : Incremento que se produce en el incremento del valor esperado variable J por incremento de P cuando se produce un incremento unitario de R.

Para la estimación de este nuevo modelo para los 4 lenguajes a la vez se ha requerido un total de 4 grados de libertad en la estimación de los parámetros y el resto han sido para el residuo. Una vez estimado el modelo y mediante el algoritmo de optimización explicado en la sección A.1 se calculan los valores de las variables P y R que minimizan la función objetivo que aparece en la ecuación A.23 bajo las restricciones:

- $0{,}0 \leq P \leq 1{,}0$
- $0{,}0 \leq R \leq 1{,}0$

A.4. Cálculo de intervalos de confianza.

En los diferentes capítulos de esta tesis ha sido necesario el cálculo de intervalos de confianza para poder evaluar el efecto de las diferentes propuestas que se han ido realizando a nivel de significación estadística. A continuación se muestran los métodos utilizados para los cálculos de los diferentes intervalos.

A.4.1. Intervalo de confianza en una población normal.

Sea n el número de muestras, $\bar{x}$ la media muestral y S_{n-1} la cuasidesviación típica muestral. En la ecuación A.24 se observa el cálculo del intervalo de confianza en poblaciones normales para la media poblacional y con una confianza $1-\alpha$.

$$\left[\bar{x} - t_{n-1}^{\frac{\alpha}{2}} \frac{S_{n-1}}{\sqrt{N}}; \bar{x} + t_{n-1}^{\frac{\alpha}{2}} \frac{S_{n-1}}{\sqrt{N}}\right] \tag{A.24}$$

A.4.2. Intervalo de confianza en una población binomial.

Sea X una variable discreta que se distribuye conforme a una distribución $B(n,p)$, donde n es el tamaño de la muestra y p la probabilidad del suceso buscado. En condiciones bastantes generales dicha distribución se puede aproximar a $N(np, \sqrt{npq})$. A partir de aquí, si en lugar de calcular la frecuencia absoluta se calcula la frecuencia relativa $\hat{p} = \dfrac{X}{n} \sim N(p, \sqrt{\dfrac{pq}{n}})$. Tipificando $Z = \dfrac{\hat{p}-p}{\sqrt{\dfrac{\hat{p}(1-\hat{p})}{n}}}$. La ecuación A.25 muestra el cálculo de un intervalo de confianza para una proporción por aproximación a una distribución normal con una confianza $1-\alpha$.

$$\left[\hat{p} - z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}; \hat{p} + z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right] \tag{A.25}$$

A.5. Riesgo, potencia, especificidad, precisión y curva ROC

En el capítulo 5 se ha planteado un problema de control estadístico de calidad a la hora de establecer un umbral que permita discernir las cadenas correctas de las no correctas. En este apartado se establecen las relaciones existentes entre diferentes conceptos como Riesgo, potencia, especificidad, precisión y curva ROC.

En inferencia estadística, siempre se parte de dos hipótesis complementarias, H_0 y H_1, donde H_0, suele ser una hipótesis conservadora que representará la idea de que nada ha cambiado en el proceso que se está evaluando mientras que H_1 representará lo contrario de la primera. A partir de aquí, en inferencia se realizará un contraste de hipótesis, unilateral o bilateral según convenga en cada caso, en el caso propuesto en el capítulo 5 estaríamos hablando de un contraste unilateral superior. Para poder contrastar las hipótesis se hará uso de las evidencias medibles en un momento dado con lo que o bien se aceptará H_0 o bien se rechazará y se admitirá como válida H_1.

A modo de ejemplo, imaginemos un proceso en el que necesitamos controlar la calidad en la producción de determinadas piezas mecánicas. Para ello, vamos a plantear un control estadístico de calidad basado en el muestreo y lo primero que se debe de hacer es definir dicho parámetro de calidad, la naturaleza del cual ha de ser numérica. Una vez definido dicho parámetro vamos a considerar que nuestro proceso de producción es correcto si se puede considerar que la media poblacional de dicho parámetro toma un valor menor o igual que x y se considerará un caso unilateral en el que nuestro proceso de producción es incorrecto si el valor de la media poblacional es mayor que x. A partir de aquí definiremos H_0 y H_1 como:

$$H_0\text{: } \mu \leq x$$
$$H_1\text{: } \mu > x$$

Para poder decidir entre H_0 o H_1 el siguiente paso será obtener n muestras de la pieza y medir el parámetro de calidad en cuestión en cada una de ellas, a partir de estas mediciones se calculan los parámetros muestrales media muestral, $\bar{x}$, y cuasidesviación típica muestral, s_{n-1} y en este caso concreto se hará uso de la distribución t-Student, para decidir sobre la aceptación o rechazo de la hipótesis H_0, puesto que $\dfrac{\bar{x}-\mu}{\frac{s_{n-1}}{\sqrt{n}}} \sim t_{n-1}$.

Una vez identificada la distribución de probabilidad, en nuestro caso t_{n-1}, ya podemos establecer sobre ésta las zonas de aceptación y rechazo de la hipótesis H_0. Para ello, debemos de asumir un riesgo de tipo I (α) a partir del cual podremos dibujar, sobre la distribución de probabilidad t_{n-1} el área que supone la aceptación de la hipótesis H_0 y que tendrá una probabilidad de $1-\alpha$, y la zona de rechazo de la hipótesis H_0, que en el caso que nos ocupa, al tratarse de un contraste unilateral superior será la cola del extremo derecho cuya masa de probabilidad acumulada representará al riesgo α asumido previamente. Desde el punto de vista de H_1, nos encontramos con un error de tipo II (β) que será la probabilidad de decidir, a partir de las evidencias, que la hipótesis H_0 es la correcta cuando realmente es H_1 la correcta y tendremos una probabilidad $1-\beta$, a la que vamos a denominar como *potencia*, de decidir que H_1 es la hipótesis correcta cuando realmente lo es.

Así pues, la potencia de un experimento estará relacionada con la capacidad de detectar H_1 como válida cuando realmente lo es. Esta potencia, está muy relacionada con el grado de separabilidad de las distribuciones de piezas correctas e incorrectas y en un control de calidad basado en el muestreo, como el comentado aquí a modo de ejemplo, dicha separabilidad estará relacionada con la cantidad de evidencias, n, de que se disponga en un momento dado y la s_{n-1} de la muestra, de tal manera que cuanto mayor sea el número de evidencias y

menor sea el valor de s_{n-1} mayor será la probabilidad $1-\beta$. Por todo ello, el establecimiento previo de la potencia deseada en un experimento es de gran utilidad a la hora de establecer el tamaño de la muestra. A continuación y a modo de resumen mostramos la tabla A.3 sobre lo comentado anteriormente.

Tabla A.3: Probabilidades de aciertos y errores

	H_0 (real)	H_1 (real)
$\hat{H}_0$ (estimada)	P($\hat{H}_0$|H_0)=$1-\alpha$	P($\hat{H}_0$|H_1)=β
$\hat{H}_1$ (estimada)	P($\hat{H}_1$|H_0)=α	P($\hat{H}_1$|H_1)=$1-\beta$

La tabla A.3 tiene una relación directa con la tabla de clasificación de los tipos de aciertos y errores que se muestra en A.4. Cabe destacar que en la tesis presentada se ha considerado VP a toda cadena clasificada como correcta que realmente es correcta y como VN a toda cadena clasificada como incorrecta que realmente es incorrecta, asociando además la idea de correcto a la hipótesis H_0 y la idea de incorrecto a la hipótesis H_1.

Tabla A.4: Tabla cruzada de clasificación de aciertos y errores

	H_0 (real)	H_1 (real)
$\hat{H}_0$ (estimada)	*Verdadero Positivo (VP)*	*Falso Positivo (FP)*
$\hat{H}_1$ (estimada)	*Falso Negativo (FN)*	*Verdadero Negativo (VN)*

A partir de estos conceptos se definen:

Sensibilidad (S): Capacidad de detectar que el proceso industrial está produciendo correctamente desde el punto de vista del parámetro de calidad. Se calcula a partir de la fórmula:

$$S = \frac{VP}{VP+FN} = 1-\alpha$$

Especificidad (E): Capacidad de detectar que el proceso industrial está produciendo de manera incorrecta desde el punto de vista del parámetro de calidad. Se calcula a partir de la fórmula:

$$E = \frac{VN}{VN+FP} = 1-\beta$$

Precisión (P)(*Accuracy*): Es la proporción de resultados correctos en la población. Se calcula a partir de la fórmula:

$$P = \frac{VP+VN}{VP+VN+FP+FN}$$

donde si se dispone de la probabilidad a priori de tratarse de una producción correcta

$$p_c = \frac{VP+FN}{VP+VN+FP+FN}$$

entonces:

$$P = p_c(1-\alpha) + (1-p_c)(1-\beta)$$

Curva ROC (ROC): Es una gráfica en el que las abcisas resultan de restarle a 1 la especificidad (1-E) y las ordenadas coinciden con la sensibilidad (S). De manera equivalente es ver como se incrementa el valor $1-\alpha$ en las ordenadas conforme se incrementa el riesgo de segunda especie β en el eje de abcisas.

A.6. Justificación e interpretación del modelo loglineal planteado

El modelo matemático que subyace en los autómatas compuestos de los capítulos 4 y 6 sobre los que se busca la cadena con menor coste de transformación es un modelo loglineal cuyos parámetros vamos a analizar aquí. Así pues, dado el modelo:

$$Y = X_1^{\lambda_1} X_2^{\lambda_2} \dots X_i^{\lambda_i} \dots X_n^{\lambda n} \tag{A.26}$$

Donde en este caso λ_i nos mide la elasticidad de la variable X_i. Para interpretar el significado de los parámetros se debe calcular el cociente de los promedios de las variables explicativas en dos instantes, t y $t+1$.

$$\frac{E(Y|X_{i_{t+1}})}{E(Y|X_{i_t})} = \frac{X_{1_t}^{\lambda_1} X_{2_t}^{\lambda_2} \dots X_{i_{t+1}}^{\lambda_i} \dots X_{n_t}^{\lambda_n}}{X_{1_t}^{\lambda_1} X_{2_t}^{\lambda_2} \dots X_{i_t}^{\lambda_i} \dots X_{n_t}^{\lambda_n}} = \left(\frac{X_{i_{t+1}}}{X_{i_t}}\right)^{\lambda_i} \tag{A.27}$$

sacando logaritmos,

$$\log\left(\frac{E(Y|X_{i_{t+1}})}{E(Y|X_{i_t})}\right) = \log\left(\left(\frac{X_{i_{t+1}}}{X_{i_t}}\right)^{\lambda_i}\right) \tag{A.28}$$

que es equivalente a

$$E\left(\log(Y|X_{i_{t+1}}) - \log(Y|X_{i_t})\right) = \lambda_i \left(\log\left(X_{i_{t+1}}\right) - \log\left(X_{i_t}\right)\right) \tag{A.29}$$

y despejando λ_i

$$\lambda_i = \frac{E\left(\log(Y|X_{i_{t+1}}) - \log(Y|X_{i_t})\right)}{\log\left(X_{i_{t+1}}\right) - \log\left(X_{i_t}\right)} \tag{A.30}$$

lo que equivaldría al incremento porcentual del valor esperado de Y, $E(Y)$, cuando la variable explicativa X_i aumenta en un uno por cien y el resto de variables mantienen sus valores. Así pues cuanto mayor sea este parámetro, respecto del resto de parámetros presentes en el modelo, mayor será la importancia de la información del submodelo asociado a λ_i sobre el modelo general. En caso de un λ_i negativo equivaldría al decremento porcentual de E(Y), cuando la variable explicativa X_i aumenta en un uno por cien manteniéndose constantes el resto de variables explicativas en el modelo. Esto desde un punto de vista probabilístico supondría que conforme más probable fuese el submodelo asociado al parámetro λ_i menos probable sería el modelo general, por lo que el algoritmo de búsqueda del camino más probable tendería a ofrecer los caminos en los que este submodelo ha tenido una probabilidad más baja.

El modelo A.26 se puede expresar también como un modelo loglineal de la siguiente forma:

$$\log Y = \lambda_1 \log X_1 + \lambda_2 \log X_2 + \cdots + \lambda_i \log X_i + \cdots + \lambda_n \log X_n \tag{A.31}$$

El modelo que subyace tras la composición de diversas fuentes de información tal y como se plantea arriba, puede ser modelado como un modelo loglineal idéntico al planteado en A.26.

$$\pi_M = \pi_H^{\lambda_h} \pi_E{}^{\lambda_e} \pi_L^{\lambda_l} \tag{A.32}$$

$$\log \pi_M = \lambda_h \log \pi_H + \lambda_e \log \pi_E + \lambda_l \log \pi_L \tag{A.33}$$

donde

π_H : Probabilidad de una arista en el modelo de hipótesis(H).

π_E : Probabilidad de una arista en el modelo de error(E).

π_L : Probabilidad de una arista en el modelo de lenguaje(L).

π_M : Probabilidad de una arista en un modelo completo a partir de la composición independiente de los modelos de hipótesis(H), error(E) y de lenguaje(L).

A partir de esto vamos a estudiar el efecto que los parámetros y las probabilidades tienen sobre el modelo.

$\pi_H = 1 \rightarrow \pi_M = \pi_E^{\lambda_e} \pi_L^{\lambda_l} \Rightarrow$ por lo que la variabilidad de la probabilidad en la arista del modelo la decidirá el modelo de error (E) y el modelo de lenguaje (L).

$\pi_H = 0 \rightarrow \pi_M = 0 \Rightarrow$ si en alguno de los modelos la arista es imposible entonces será una arista imposible para el modelo general.

Imaginemos ahora que $0 < \pi_H < 1$, entonces

$\lambda_h = 0 \rightarrow \pi_H^{\lambda_h} = 1 \rightarrow \pi_M = \pi_E^{\lambda_e} \pi_L^{\lambda_l} \Rightarrow$ por lo que la variabilidad de la probabilidad en cada una de las aristas del modelo la decidirán el modelo de error (E) y el modelo de lenguaje (L).

$\lambda_h = \infty \rightarrow \pi_H^{\infty} = 0 \rightarrow \pi_M = 0 \Rightarrow$ las aristas se convierten en imposibles en el modelo general y la probabilidad de éste tenderá a 0.

$\lambda_h = -\infty \rightarrow \pi_H^{-\infty} = \dfrac{1}{0} = \infty \rightarrow \pi_M = \infty \Rightarrow$ el valor de las aristas acaba siendo gobernado por el modelo π_H.

Cabe destacar que si $0 \leq \lambda_h \leq \infty$ entonces $0 \leq \pi_H^{\lambda_h} \leq 1$ y que si $-\infty \leq \lambda_h \leq 0$ entonces $0 \leq \pi_H^{\lambda_h} \leq \infty$, pudiendo producir este último un modelo no acotado superiormente.

BIBLIOGRAFÍA

Al Azawi, M. I. A. and Breuel, T. M. (2014). Context-dependent confusions rules for building error model using weighted finite state transducers for OCR post-processing. In *11th IAPR International Workshop on Document Analysis Systems, DAS 2014, Tours, France, April 7-10, 2014*, pages 116–120.

Alabau, V., Martínez-Hinarejos, C. D., Romero, V., and Lagarda, A. L. (2014). An iterative multimodal framework for the transcription of handwritten historical documents. *Pattern Recognition Letters*, 35:195–203. Frontiers in Handwriting Processing.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). Openfst: A general and efficient weighted finite-state transducer library. In *In Proceedings of CIAA, LNCS 4783*, pages 11–23.

Amengual, J.-C. and Vidal, E. (1998). Efficient error-correcting viterbi parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(10):1109–1116.

Arlandis, J., Perez-Cortes, J., and Cano, J. (2002). Rejection strategies and confidence measures for a k-NN classifier in an OCR task. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 576 – 579 vol.1.

Arlandis, J., Pérez-Cortes, J. C., Navarro-Cerdan, J. R., and Llobet, R. (2010). Rejection threshold estimation for an unknown language model in an ocr task. In *SSPR/SPR*, pages 738–747. Springer.

Bastide, R., Navarre, D., Palanque, P., Schyn, A., and Dragicevic, P. (2004). A model-based approach for real-time embedded multimodal systems in military aircrafts. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 243–250, New York, NY, USA. ACM.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300.

Berghel, H. L. (1987). A logical framework for the correction of spelling errors in electronic documents. *Inf. Process. Manage.*, 23:477–494.

Bertolami, R., Zimmermann, M., and Bunke, H. (2006). Rejection strategies for offline handwritten text line recognition. *Pattern Recognition Letters*, 27(16):2005–2012.

Broadwater, J. and Chellappa, R. (2010). Adaptive threshold estimation via extreme value theory. *IEEE Transactions on Signal Processing*, 58:490–500.

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

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

Culik, I. K. and Kari, J. (1997). *Digital images and formal languages*, pages 599–616. Springer-Verlag New York, Inc., New York, NY, USA.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, 1(1):269–271.

Eisner, J. (2002). Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–8, Philadelphia.

Evans, M. and Rosenthal, J. (2005). *Probabilidad y estadística*. Reverté.

Farooq, F., Jose, D., and Govindaraju, V. (2009). Phrase-based correction model for improving handwriting recognition accuracies. *Pattern Recogn.*, 42:3271–3277.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874.

Fernando, M. M., Pereira, O., and Riley, M. (1996). Weighted automata in text and speech processing. In *In ECAI-96 Workshop*, pages 46–50. John Wiley and Sons.

Gandrabur, S., Foster, G. F., and Lapalme, G. (2006). Confidence estimation for nlp applications. *TSLP*, 3(3):1–29.

Garay-Vitoria, N. and Abascal, J. (2006). Text prediction systems: A survey. *Univers. Access Inf. Soc.*, 4(3):188–203.

Garay-Vitoria, N. and Abascal, J. (2010). Modelling text prediction systems in low- and high-inflected languages. *Computer Speech & Language*, 24(2):117–135.

Garcia, P. and Vidal, E. (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:920–925.

Hall, P. A. V. and Dowling, G. R. (1980). Approximate string matching. *ACM Comput. Surv.*, 12:381–402.

Hanczar, B. and Dougherty, E. R. (2008). Classification with reject option in gene expression data. *Bioinformatics (Oxford, England)*, 24(17):1889–1895.

Hansen, B. E. (2000). Sample splitting and threshold estimation. *Econometrica*, 68(3):575–604.

Hassan Awadallah, A., Noeman, S., and Hassan, H. (2008). Language independent text correction using finite state automata. In *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, pages 913–918.

He, C. L., Lam, L., and Suen, C. Y. (2009). A novel rejection measurement in handwritten numeral recognition based on linear discriminant analysis. In *10th International Conference on Document Analysis and Recognition*, pages 451–455. IEEE Computer Society.

Hull, J. J. and Srihari, S. N. (1982). Experiments in text recognition with binary n-gram and viterbi algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4(5):520–530.

Jelinek, F. (1991). Up from trigrams! The struggle for improved language models. In *Proceedings European Conference on Speech Communication and Technology*, pages 1037–1039.

Kae, A., Huang, G. B., and Learned-Miller, E. G. (2009). Bounding the probability of error for high precision recognition. *CoRR*, abs/0907.0418.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28 – 44.

Köksal, G., Batmaz, I., and Testik, M. C. (2011). Review: A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, 38(10):13448–13467.

Kolak, O. and Resnik, P. (2005). Ocr post-processing for low density languages. In *Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT)*, pages 867–874. Association for Computational Linguistics.

Kuich, W. and Salomaa, A. (1986). *Semirings, automata, languages*. Springer-Verlag, London, UK.

Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439.

Landgrebe, T., Paclík, P., and Duin, R. P. W. (2006). Precision-recall operating characteristic (p-roc) curves in imprecise environments. In *International Conference on Pattern Recognition ICPR (4)*, pages 123–127.

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

Li, M. and Sethi, I. K. (2006a). Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1251–1261.

Li, M. and Sethi, I. K. (2006b). Confidence-based classifier design. *Pattern Recognition*, 39(7):1230–1240.

Lindberg, J., Koolwaaij, J., Hutter, H., Genoud, D., Pierrot, J., Blomberg, M., and Bimbot, F. (1998). Techniques for a priori decision threshold estimation in speaker verification. In *Proceedings RLA2C*, pages 89–92.

Llobet, R., Navarro-Cerdan, J. R., Perez-Cortes, J.-C., and Arlandis, J. (2010a). Efficient ocr post-processing combining language, hypothesis and error models. In *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, pages 728–737, Berlin, Heidelberg. Springer-Verlag.

Llobet, R., Navarro-Cerdán, J.-R., Pérez-Cortes, J. C., and Arlandis, J. (2010b). Ocr post-processing using weighted finite-state transducers. In *ICPR*, pages 2021–2024.

Meyer, G. F., Mulligan, J. B., and Wuerger, S. M. (2004). Continuous audio-visual digit recognition using n-best decision fusion. *Information Fusion*, 5(2):91–101.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Comput. Linguist.*, 23:269–311.

Mohri, M. (2004). *Weighted Finite-State Transducer Algorithms An Overview*. Physica-Verlag.

Mohri, M., Pereira, F., and Riley, M. (2000). The design principles of a weighted finite-state transducer library. *THEORETICAL COMPUTER SCIENCE*, 231:17–32.

Müller, C. and Weinberg, G. (2011). Multimodal input in the car, today and tomorrow. *IEEE MultiMedia*, 18(1):98–103.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7:308–313.

Neuhoff, D. (1975). The viterbi algorithm as an aid in text recognition. *IEEE Transactions on Information Theory*.

Ozturk, A., Chakravarthi, P. R., and Weiner, D. D. (1996). On determining the radar threshold for non-gaussian processes from experimental data. *IEEE Transactions on Information Theory*, 42(4):1310–1316.

Paladini, E. P. (2000). An expert system approach to quality control. *Expert Systems with Applications*, 18(2):133 – 151.

Park, Y. A. and Levy, R. (2011). Automated whole sentence grammar correction using a noisy channel model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 934–944, Stroudsburg, PA, USA. Association for Computational Linguistics.

Perez-Cortes, J., Amengual, J., Arlandis, J., and Llobet, R. (2000). Stochastic error correcting parsing for ocr post-processing. In *International Conference on Pattern Recognition ICPR-2000*, volume 4, pages 405–408, Barcelona, (Spain).

Pérez-Cortes, J. C., Llobet, R., and Arlandis, J. (2000). Fast and accurate handwritten character recognition using approximate nearest neighbours search on large databases. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 767–776, London, UK. Springer-Verlag.

Perez-Cortes, J.-C., Llobet, R., Navarro-Cerdan, J. R., and Arlandis, J. (2011). Improvement of embedded human-machine interfaces combining language, hypothesis and error models. *2012 23rd International Workshop on Database and Expert Systems Applications*, 0:359–363.

Pitrelli, J. F., Subrahmonia, J., and Perrone, M. P. (2006). Confidence modeling for handwriting recognition: algorithms and applications. *International Journal of Document Analysis*, 8(1):35–46.

Raman, K., Swaminathan, A., Gehrke, J., and Joachims, T. (2013). Beyond myopic inference in big data pipelines. In Dhillon, I. S., Koren, Y., Ghani, R., Senator, T. E., Bradley, P., Parekh, R., He, J., Grossman, R. L., and Uthurusamy, R., editors, *KDD*, pages 86–94. ACM.

Rijsbergen, C. (1979). *Information retrieval*. Butterworths.

Riley, M., Pereira, F., and Mohri, M. (1997). Transducer composition for context-dependent network expansion. In *In Proceedings of Eurospeech'97. Rhodes*, pages 1427–1430.

Schlapbach, A., Wettstein, F., and Bunke, H. (2008). Estimating the readability of handwritten text - a support vector regression based approach. In *19th. International Conference on Pattern Recognition*, pages 1–4.

Serrano, N., Civera, J., Sanchis, A., and Juan, A. (2014). Effective balancing error and user effort in interactive handwriting recognition. *Pattern Recognition Letters*, 37(0):135 – 142.

Suhm, B., Myers, B., and Waibel, A. (1999). Model-based and empirical evaluation of multimodal interactive error correction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 584–591, New York, NY, USA. ACM.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293.

T Breuel (1994). Language Modeling for a Real-world Handwriting Recognition Task. In *AISB Workshop on Computational Linguistics for Speech and Handwriting Recognition, 1994*.

Taghva, K. and Stofsky, E. (2001). Ocrspell: an interactive spelling correction system for ocr errors in text. *International Journal of Document Analysis and Recognition*, 3:2001.

Tong, X. and Evans, D. A. (1996). A statistical approach to automatic ocr error correction in context. In *Proceedings of the Fourth Workshop on Very Large Corpora (WVLC-4*, pages 88–100.

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

Ueffing, N. and Ney, H. (2007). Word-level confidence estimation for machine translation. *Comput. Linguist.*, 33:9–40.

Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., and Carrasco, R. C. (2005). Probabilistic finite-state machines-part ii. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1026–1039.

Villafranca, R. and Ramajo, L. (2005). *MÉTODOS ESTADÍSTICOS EN INGENIERÍA*. Universidad Politécnica de Valencia.