UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Aportaciones al Reconocimiento Automático de Texto Manuscrito

BORRADOR DE TESIS

Tesis Doctoral
presentada por Moisés Pastor i Gadea
dirigida por el Dr. Enrique Vidal y el Dr. Alejandro H. Toselli

27 de abril de 2007

Aportaciones al Reconocimiento Automático de Texto Manuscrito

Moisés Pastor i Gadea

Trabajo realizado bajo la dirección de los Doctores
Dr. Enrique Vidal Ruíz y Dr. Alejandro Héctor Toselli
y presentado en la Universidad Politécnica de Valencia
para optar al grado de Doctor en Informática

Valencia, 27 de abril de 2007

A mon pare, que ja no es conscient del que açò significa.

AGRADECIMIENTOS

Quiero agradecer a todos aquellos que de una manera u otra han tenido que ver con la realización del presente trabajo.

Estoy agradecido a los componentes del grupo PRHLT que me han proporcionado un entorno estimulante, especialmente a Alberto Sanchis, como amigo y experto en verificación de hipótesis, por su orientación y por haber estado a deshoras contandome los entresijos de la verificación. A Alejandro H. Toselli, amigo y codirector de este trabajo, que ha ayudado a que esta tesis tomara la linea de trabajo que ha tomado. Gracias a Carlos Martínez por su compañerismo y apoyo, y a Verónica por su ayuda en la experimentación. A la gente del ITI, la que está y la que ya no, a Paco Casacuberta, a Alfons, y muy especialmente a Enrique Vidal por todo.

A ATROS con el que he compartido muchas horas de angustias, alegrías y decepciones, y con el que he aprendido a programar.

A Clara por su paciencia e incondicionalidad, y por brindarme el equilibrio y el sosiego que me han permitido avanzar en la consecución de esta tesis, por cuidarme cuando he estando enclaustrado, por forzarme a descansar cuando me he ofuscado, y por todas esas pequeñas cosas que no sería propio dejar constancia aquí.

A todos mi más sincera gratitud.

Moisés Pastor i Gadea
Valencia, 27 de abril de 2007

RESUMEN

En esta tesis se estudia el problema de la robustez en los sistemas de *reconocimiento automático de texto manuscrito off-line*. Los sistemas de reconocimiento automático de texto manuscrito estarán maduros para su uso generalizado, cuando sean capaces de ofrecer a cualquier usuario, sin ningún tipo de preparación o adiestramiento para su utilización, una productividad razonable. Se hace necesario pues, construir sistemas flexibles y robustos en cuanto a la entrada, de tal manera que no se requiera del escritor ningún esfuerzo extra, que no haría si escribiese para ser leído por un humano.

La intención del preproceso de la señal es hacer el sistema invariante a fuentes de variabilidad que no ayuden a la clasificación. En la actualidad no hay definida una solución general para conseguir invariabilidad al estilo de escritura, y cada sistema desarrolla la suya *ad-hoc*. En esta tesis se explorarán diferentes métodos de normalización de la señal de entrada *off-line*. Para ello se hace un amplio estudio de algoritmos de preproceso, tanto a nivel de toda la imagen: umbralización, reducción del ruido y corrección del desencuadre; como a nivel de texto: *slope*, *slant* y normalización del tamaño de los caracteres.

Los sistemas dependientes del escritor consiguen mejores tasas de acierto que los independientes del escritor. Por otra parte, los sistemas independientes del escritor tienen más facilidad para reunir muestras de entrenamiento. En esta tesis se estudiará la adaptación de sistemas independientes del escritor para su utilización por un único escritor, con la intención de que a partir de una pocas muestras producidas por este escritor se mejore la productividad del sistema (para este escritor), o lo que es lo mismo, que éste pueda escribir de manera más relajada sin que el sistema pierda productividad.

Los sistemas *de reconocimiento de texto manuscrito* no están exentos de errores. No sólo interesa saber el número de errores que producirá el sistema, sino que es importante saber qué unidades de la hipótesis producida por el reconocedor están mal reconocidas, o no se tiene garantía de que estén bien reconocidas, para así poderlas corregir manualmente. En esta tesis se estudia la adaptación de las técnicas de verificación de hipótesis más exitosas en el campo del *reconocimiento automático del habla* para ser usadas por los sistemas de *reconocimiento de texto manuscrito*.

El problema del *reconocimiento automático del habla continua* y el del *reconocimiento automático de texto manuscrito* presentan grandes similitudes. Debido a ello, se ha adaptado el reconocedor del habla continua ATROS (*Automaticaly Trainable Recognizer Of Speech*) [PSCV01] para ser utilizado como reconocedor automático de texto manuscrito. El reconocedor ATROS es un reconocedor basado

en modelos sin segmentación explícita, HMMs.

RESUM

A aquesta tesi s'estudia el problema de la robustesa dels sistemes de *reconeixement automàtic de text manuscrit off-line*. Aquestos sistemes estaran madurs per a la seua utilització generalitzada, quant siguen capaços d'oferir a qualsevol usuari, sense cap tipus de preparació o adestrament per a la seua utilització, una productivitat raonable. Es fa necessari doncs, construir sistemes flexibles i robustos en tant a l'entrada, de tal manera que no es requereisca de l'escriptor cap esforç extra, que no faria si escrigués per a ser llegit pe un humà.

La intenció del preprocés de la senyal és fer el sistema invariant a les fonts de variabilitat que no ajuden a la classificació. En la actualitat no hi ha definida una solució general per aconseguir invariabilitat a l'estil d'escriptura i cada sistema desenvolupa la seua *ad-hoc*. A aquesta tesi s'exploraran diversos mètodes de normalització del senyal d'entrada *off-line*. Per això es farà un ampli estudi de algorismes de preprocés, tant a nivell de tota la imatge: umbralització, reducció del soroll i correcció del desencuadre; com a nivell de text: *slope*, *slant* i normalització del tamany dels caracters.

Els sistemes dependents de l'escriptor aconsegueixen millors taxes d'encert que els independents de l'escriptor. Per una altra banda, els sistemes independents de l'escriptor tenen més facilitat per a reunir mostres d'entrenament. A aquesta tesi s'estudiarà la adaptació de sistemes independents de l'escriptor per a ser emprats per un únic escriptor, amb la intenció de que a partir d'unes poques mostres produïdes per eixe escriptor es millore la productivitat del sistema (per a eixe escriptor), o el que és el mateix, que aquest puga escriure de manera més relaxada sense que el sistema perda productivitat.

Els sistemes de *reconeixement automàtic de text manuscrit* no estan exents d'errors. No sols interessa saber el nombre d'errors que produirà el sistema, a més és important saber quines unitats de la hipòtesi produïda pel reconeixedor estan mal reconegudes, o no es té garanties de que estiguen ben reconegudes, per poderles corregir manualment. A aquesta tesi s'estudia la adaptació de les tècniques de verificació d'hipòtesis més exitoses al camp de *reconeixement automàtic de la parla*.

El problema del *reconeixement automàtic de la parla continua* y el del *reconeixement automàtic de text manuscrit* presenten grans similituds, degut a lo qual, s'ha adaptat el reconeixedor automàtic de la parla continua ATROS (*Automaticaly Trainable Recognizer Of Speech*) [PSCV01] per a ser emprat com a reconeixedor automàtic de text manuscrit. El reconeixedor ATROS és un sistema basat en models sense segmentació prèvia, HMMs.

ABSTRACT

This thesis is addressed to the subject of robustness on automatic handwritten text recognizers. These systems will be available for a generalized use when them can provide to any user, without any specific training, a reasonable productivity. These systems must not require any effort that a writer do not if he/her are writing for a human being. Then, it is needed to build robust and flexible systems from the input point of view.

The signal prepprocess aims to make the system invariant to all those sources that do not help to classify the handwriting text. Nowadays, there are not a standard solution for achieve invariability to the handwritten style. Every system has its own ad-hock solution. This thesis explores several methods for off-line input signal normalization. For that, a prepprocess algorithms spread study is made. The algorithms are classified as page level: threshold, noise reduction and skew angle correction; and text level: *slope* and *slant* angle correction, and character size normalization.

Writer dependent systems achieve consistently better recognition results compared to writer independent systems. On the other hand, collecting a large number of data samples for training writer independent systems is easier than the collecting a large data samples from a single writer. In this work we made a study of independent systems adaptation for be used by a single writer. This way, the writer would writes in a more relaxed way without system productivity loss.

Automatic handwritten recognition systems are not extent of errors. On the other hand, it is interesting, not only know the number of errors but know what hypothesis units are errors, or are suspected to be bad classified, in order to correct them manually. At this thesis, a successful speech recognition hypothesis verification techniques are adapted to be used on *automatic manuscript text recognition* systems.

The *automatic continuous speech recognition* problem have important similarities with the *manuscript text automatic recognition* one. Because that, the speech recognition engine ATROS (*Automatically Trainable Recognizer Of Speech*) [PSCV01] will be adapted to be used as manuscript text recognizer. ATROS is a recognizer engine based on free segmentation models, hidden Markov models.

PROLOGO

El contenido de esta tesis es el resultado del trabajo realizado por el autor en el campo de la robustez de los sistemas de reconocimiento automático de texto manuscrito (RATM). Este trabajo se enmarca dentro de una de las lineas de investigación abiertas dentro del grupo de investigación *Pattern Recognition and Human Lenguage Technology*, PRHLT[1] del *Departament de Sistemes Informàtics i Computació de València*.

El RATM es un campo que ha crecido en interés por parte de la comunidad científica y por la industria en las últimas dos décadas. Los avances en este campo han sido espectaculares y han ayudado a agilizar los envíos postales; a tratar automáticamente los cheques bancarios; o a verificar firmas, por poner un ejemplo.

En el mercado se pueden encontrar productos basados en la tecnología OCR (*Optical Character Recognition*) con altas tasas de acierto. Estos productos se utilizan sobre texto impreso, que es muy homogéneo respecto al tipo y forma de los caracteres, además de permitir una segmentación relativamente fácil. El problema de la segmentación se hace muy patente cuando se intenta utilizar la tecnología OCR con texto manuscrito donde es usual que los caracteres estén conectados. El texto manuscrito presenta muchos más grados de libertad que el texto impreso: diferentes estilos de escritura, cursiva, irregularidad en las lineas base de las frases, diferentes tipos de instrumentos de escritura, solapamientos, etc. que hacen que el éxito de los sistemas OCR se diluya cuando se intentan aplicar a texto manuscrito.

Los sistemas actuales RATM ofrecen prestaciones de manera aceptable sólo en el caso de que se apliquen en dominios semánticos muy restringidos, o en aquellos casos donde la talla del vocabulario es extremadamente pequeña, de unas pocas decenas de palabras, o para sistemas dependientes del escritor, los cuales requieren un gran esfuerzo, previo a su utilización, por parte del escritor.

A pesar del gran avance producido en el campo de los sistemas RATM, es necesario un gran esfuerzo para lograr un grado suficiente de madurez que permita la utilización a gran escala de los sistemas RATM. Se necesita poder trabajar con tallas de vocabulario grandes, del orden de varios centenares de miles de palabras. También es necesario conseguir mucha más tolerancia al estilo de escritura, para que cualquiera, sin ningún tipo de preparación previa, pueda utilizar el sistema.

Esta tesis está ”organizada” en ocho capítulos. En el primero se expone una introducción de los sistemas RATM y del contexto en el que se engloban. En este capítulo se puede encontrar una clasificación de los sistemas de RATM con respecto al módulo de adquisición, al de parametrización, con respecto al módulo de clasificación/reconocimiento, y con respecto a módulo de entrenamiento. También

[1] http:/prhlt.iti.es

se incluye una descripción del estado del arte y una relación de sistemas comerciales basados en RATM.

En el capítulo 2 se exponen los fundamentos teóricos necesarios para comprender el funcionamiento básico de los sistemas RATM utilizados a lo largo del presente trabajo. En este capítulo se revisa la teoría y los algoritmos aplicados a modelos ocultos de Markov HMM y a los modelos de lenguaje, haciendo especial hincapié en los modelos de n-gramas. En la última sección, se expone la métrica utilizada para medir la productividad de los sistemas.

En el capítulo 3 se exponen los detalles de los corpus utilizados para entrenar y probar las técnicas estudiadas.

En el capítulo 4 se exponen las técnicas de preproceso de imagen más usadas en el campo del RATM. Así mismo, se han propuesto mejoras para algunos métodos, y se han introducido nuevos. Los métodos de normalización de la imagen se han estructurado en dos grandes grupos: los que normalizan el texto a nivel de página, y los que lo hacen centrándose en el texto contenido. En el primer grupo se han estudiado técnicas globales y adaptativas de umbralización de la imagen, métodos de reducción del ruido, y de corrección del desencuadre o *skew*. En el segundo grupo se han estudiado técnicas de corrección del ángulo de *slant*, y de normalización del tamaño del texto.

El capítulo 5 trata la extracción de características para RATM, donde se puede encontrar una breve revisión de las características más usadas en este campo, para luego centrarse en aquellas utilizadas en este trajo.

En el capítulo 6, tras una breve introducción de los métodos utilizados de adaptación al escritor, se presenta el trabajo realizado para adaptar estos métodos, originarios del reconocimiento automático del habla, al dominio del RATM.

En el capítulo 7 se exponen dos técnicas de verificación aplicadas con éxito en el dominio del reconocimiento automático de voz y su adaptación al dominio del reconocimiento de texto manuscrito.

En el capítulo 8 se exponen las conclusiones y ampliaciones del trabajo para un futuro.

Moisés Pastor i Gadea
Valencia, 27 de abril de 2007

ÍNDICE GENERAL

CAPÍTULO 1

INTRODUCCIÓN

A lo largo de la última mitad del siglo pasado se ha producido un avance considerable en los sistemas de computación automática. Hoy en día se asiste a una verdadera revolución digital. Los computadores se han convertido en elementos cotidianos, de tal manera, que estamos habituados a vivir rodeados de un sinfín de objetos controlados por computador. De echo, se asume que cualquier cosa que pueda ser codificada numéricamente es susceptible de ser manipulada utilizando computadores.

La escritura ha sido la manera más natural de almacenar y transmitir información desde la antigüedad. En nuestros días, una gran parte de la información disponible sigue estando almacenada en papel. Las bibliotecas digitales han adquirido gran relevancia, tras la aparición, y gran expansión de las redes de computadores, ya que permiten la consulta de las fuentes, de manera inmediata, aunque se esté a miles de kilómetros. Para construir bibliotecas digitales, este gran volumen de información debe ser digitalizado (convertido a algún formato numérico), y el texto convertido a formato de texto electrónico, para su almacenamiento, recuperación y fácil manipulación [WMR97].

La conservación de documentos antiguos requiere un gran cuidado porque se degradan con facilidad. La manipulación, los cambios de temperatura, de luz, etc, pueden producir daños irreparables en ellos. Este tipo de factores, que imponen una manipulación limitada, junto con la deslocalización de los mismos, son dos grandes escollos con que se encuentran los investigadores. La aparición de las bibliotecas digitales resuelve en gran parte estos problemas, porque permiten la consulta, indexación, búsquedas de palabras, etc. en los documentos disponibles, sin tener que desplazarse.

El reconocimiento automático de texto manuscrito (RATM) es el proceso por el cual se transforma de manera automática de un lenguaje simbólico, representado en un espacio bidimensional mediante grafías manuscritas, a una codificación numérica de cada una de las grafías. Esta codificación numérica es la usada internamente por los computadores para representar los caracteres. En la actualidad coexisten diferentes codificaciones como por ejemplo ASCII, UNICODE, etc.

El RATM comparte muchas características con el problema del reconocimien-

to automático del habla (RAH). Basándose en ello muchos autores han optado por utilizar tecnología del campo del RAH [Jel98, RJ93, Shu96] basada principalmente en modelos ocultos de Markov, también conocidos como HMMs (del inglés, *Hidden Markov Models*). Estos modelos se verán con detalle en el capítulo 2.

1.1. Clasificación de los sistemas de RATM.

Los sistemas de reconocimiento óptico de caracteres (OCR) (del inglés, *Optical Character Recognition*) aplicado a texto impreso, ha alcanzado un gran nivel en la últimas décadas. Gran parte de su éxito radica en la relativa facilidad para segmentar el texto en caracteres aislados, que luego son clasificados con técnicas ”maduras” como los k-vecinos más cercanos, clasificadores de Bayes o redes neuronales, por poner un ejemplo. Las prestaciones de los sistemas OCR caen drásticamente cuando se enfrentan a texto manuscrito, ya que su segmentación no es trivial.

Los sistemas RATM están compuestos por una serie de módulos, tal como se muestra en la figura 1.1. Los sistemas RATM se clasifican dependiendo de la aproximación utilizada para cada módulo. A continuación se presenta la taxonomía de los sistemas de RATM más aceptada en la literatura.

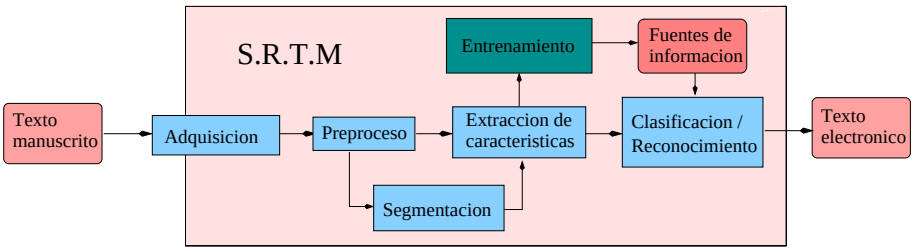


Figura 1.1: Esquema general de un reconocedor automático de texto manuscrito.

1.1.1. Taxonomía con respecto al módulo de adquisición

La percepción del mundo sensible es de naturaleza analógica. Nuestros sentidos utilizan medios de comunicación continuos, como la luz, el sonido, la presión, etc. Por otra parte, para poder tratar una imagen mediante un computador, esta debe representarse numéricamente. El módulo de adquisición se encarga representar los objetos del mundo sensible mediante números, o secuencias de números. Con respecto al módulo de adquisición, los sistemas de RATM se clasifican del siguiente modo:

- *off-line*: la adquisición se hace en dos tiempos, primero el texto se escribe en papel, o cualquier otro medio físico similar, para luego adquirir la imagen utilizando un escáner o una cámara. Para digitalizar una página, se le superpone a esta una retícula. Para cada una de las celdas de la retícula se obtiene

el valor promedio de la luz reflejada en ella. Los valores obtenidos para todas las retículas conforman una matriz, que es la representación electrónica de la imagen contenida en el papel. A cada uno de los puntos representados por esta matriz se le denomina píxel y es la unidad mínima en las imágenes electrónicas. La cuantificación de la luz de cada celda de la imagen original produce un único valor para aquellas digitalizaciones en niveles de gris, o una terna de ellos, un valor para cada color básico, para digitalizaciones en color. A la densidad de celdas por unidad de superficie se la denomina resolución. Cuanto más elevada sea la resolución, mayor realismo tendrá la imagen electrónica. De manera más formal diremos que una imagen bidimensional, $f(x, y)$, es una función, $f : \Re \times \Re \rightarrow \Re$, donde para cada punto del espacio bidimensional devuelve el nivel de luz de dicho punto. Una imagen *off-line* es una función, $I(x, y)$, donde el espacio bidimensional, y los niveles de gris se han discretizado; $I : \{1, \cdots, F\} \times \{1, \cdots, C\} \rightarrow \{1, \cdots, N\}$, N suele ser 255, y F,C son las filas y columnas de la matriz. Esta modalidad es la utilizada en esta tesis.

- *on-line*: la adquisición se realiza directamente desde algún dispositivo que registre el movimiento, con respecto a unos ejes de coordenadas, de un móvil. El adquiridor devuelve una secuencia, ordenada en el tiempo, de coordenadas. Esta secuencia constituye lo que se denomina una paramétrica, más concretamente, una curva plana o bidimensional. Una curva plana es una secuencia de puntos en un espacio bidimensional, ordenados en el tiempo; es decir una aplicación $f : I \rightarrow \Re^2$ donde $I \in]a, b[$ es un intervalo temporal. Al conjunto imagen $f(]a, b[)$ se le llama traza de la curva. Hay que tener en cuenta que una misma traza puede pertenecer a más de una curva, por ejemplo una misma imagen trazada de diferente manera tendrá una secuencia de puntos ordenada de manera distinta aunque sus trazas serán iguales. En este tipo aquisición, al igual que en el caso anterior, el espacio bidimensional y los niveles de gris se han discretizado.

1.1.2. Taxonomía con respecto al módulo de parametrización

El módulo de parametrización, también conocido como de *extracción de características*, pretende obtener aquellas características de la imagen que permitan discriminar lo mejor posible entre los patrones de cada clase, y al mismo tiempo que se elimina la información redundante. En esta fase se obtiene una representación compacta y discriminante de la señal de entrada. Las características deben ser suficientemente invariables para que estén presentes en cualquier estilo de escritura, y al mismo tiempo deben ser suficientemente discriminativas entre clases de equivalencia. La elección del tipo y número de características es una decisión crítica, y depende en gran medida de cada aplicación.

Con respecto al módulo de parametrización hay dos grandes aproximaciones clásicas:

- Aproximación no estructural o global. Cada imagen está compuesta por un objeto simple, que en futuras etapas deberá clasificarse entre un número determinado de clases. Los objetos se representan mediante un vectores de características.

- Aproximación estructural: cada patrón está compuesto de una serie de formas simples, combinadas mediante unas reglas, que rigen las relaciones estructurales. Existen dos tipos de representaciones estructurales:

  - Representación sintáctica: la salida del módulo de parametrización tiene forma de frase en algún lenguaje de descripción.
  - Representación relacional: el módulo de parametrización produce grafos relacionales, redes semánticas u otra representación que muestran las relaciones entre las primitivas.

1.1.3. Taxonomía respecto al módulo de clasificación/reconocimiento

Dependiendo de el tipo de representación de los patrones, si se trata de representaciones de objetos simples, se hablará de clasificación, mientras que si se trata de representaciones estructurales, se hablará de reconocimiento, o interpretación. En el caso de la clasificación, el patrón se clasificará entre una serie de clases de equivalencia previamente definidas. Si la aproximación es geométrica, el espacio de las características se particionará, y el objeto a clasificar se verá como un punto en ese espacio. La clasificación consiste en decidir a que partición pertenece dicho punto. Si la representación es estadística, el vector de características se verá como un vector de variables aleatorias. El proceso de clasificación consiste en construir modelos de naturaleza probabilística que minimicen la probabilidad de clasificación errónea.

Por la naturaleza del texto manuscrito, se puede decir que se trata de un problema estructural, por lo que se hablará de reconocimiento. La taxonomía más aceptada en la literatura respecto al módulo de reconocimiento se presenta a continuación:

- Aproximación basada en segmentación, o analítica: el objeto base a reconocer es el caracter. En esta aproximación, primero se segmenta el texto en caracteres que en una segunda etapa se clasificarán. Una ventaja importante de esta aproximación es que permite la construcción de sistemas con un vocabulario abierto. El principal problema es el tener que segmentar cada palabra en caracteres, lo cual no es una tarea trivial, sobretodo cuando se habla de texto manuscrito. Esta aproximación ha sido utilizada con bastante éxito en tareas de reconocimiento de texto impreso (OCR), donde la segmentación es bastante fácil. En 1973 Sayre publica un artículo [Say73] en el que aparece su famosa paradoja: ”una palabra no puede ser segmentada sin antes haber sido reconocida, y no puede ser reconocida sin antes

haber sido segmentada" con la cual parecía que se sentenciaba esta aproximación, aplicada al reconocimiento de texto manuscrito. No obstante, en esta aproximación el problema queda reducido a como conseguir una buena segmentación a nivel de caracter, puesto que para la clasificación de caracteres aislados se dispone de un amplio abanico de algoritmos consolidados [CKS95, BRST95, EYGSS99]. Los sistemas de segmentación suelen estar basados en heurísticos más o menos sofisticados [CL96, DK97]. El problema que queda abierto es el desarrollo de procedimientos automáticos de segmentación, que aprendan sus parámetros a partir de muestras de entrenamiento [Bun03].

- Aproximación global (*holistic aproach*): el objeto base a reconocer es la palabra. En esta aproximación se evita la parte difícil de la aproximación anterior, no se segmentan las palabras en caracteres, por el contrario se toma la palabra como unidad y se intenta reconocer la palabra entera. El principal problema que presenta esta aproximación es que el número de muestras necesario para estimar los modelos correctamente puede ser prohibitivo. Por otra parte, los sistemas construidos a partir de esta aproximación son robustos frente al ruido y a errores ortográficos. Es una aproximación interesante para aplicaciones con un vocabulario reducido [GS95, KGS99, MG99, MG01].

- Aproximación sin segmentación explícita (*segmentation-free*): el objeto base a reconocer es la frase. Esta aproximación intenta aprovechar las ventajas de las dos aproximaciones anteriores. Partiendo de modelos HMM a nivel de caracter, se construyen modelos HMM de cada palabra simplemente concatenando los modelos HMM de los caracteres que la conforman. Del mismo modo se construyen modelos HMM de las frases concatenando los modelos de las palabras. Al tener como modelo morfológico base el caracter, se evitan los problemas de falta de entrenamiento que sufren los modelos holísticos. Por otra parte, la inserción de nuevas palabras en el léxico es sencilla, basta con describir que modelos de caracter se enlazarán para formar el modelo de la palabra. Al reconocer la frase entera como una unidad, se evitan los problemas derivados de la segmentación, que en este caso se obtiene como subproducto del proceso de reconocimiento. Esta aproximación es la utilizada a lo largo de esta tesis.

1.1.4. Taxonomía respecto al módulo de entrenamiento

En este apartado se expone la taxonomía más extendida respecto al módulo de entrenamiento, y respecto a los modelos (fig. 1.2). Aunque en algunos pocos casos se tiene suficiente información *a priori* para generar los modelos manualmente, lo habitual es realizar un aprendizaje inductivo, donde se estiman los modelos a partir de un conjunto de muestras de ejemplo.

Dependiendo de la información que se tenga de las muestra del conjunto de entrenamiento, el aprendizaje de los modelos se puede clasificar como:

- aprendizaje supervisado: se conoce la clase a la que pertenece cada muestra. Este método de aprendizaje es el más fácil de todos, por lo que es el aprendizaje más usual.
- aprendizaje no supervisado: se utiliza en el caso de que no se sepa la clase a la que pertenece cada una de las muestras. Como cabe esperar, este es es el aprendizaje más difícil.
- con aprendizaje continuo o incremental: durante el funcionamiento normal del sistema RATM, las nuevas muestras reconocidas pasan a formar parte del corpus de entrenamiento. Periódicamente se reestimarán los modelos para que se adapten mejor al nuevo corpus.

En el caso de que el sistema sea un clasificador (los objetos son simples), los sistemas de estimación de modelos se clasifican de la siguiente manera:

- sistemas de estimación de modelos basados en teoría de la decisión: Métodos de particionamiento del espacio de representación donde muestras similares o ”cercanas” se agrupan en la misma clase [DH74, F’u99, TK03]. Los principales problemas son la definición de ”similitud” entre muestras, y la elección de las métricas apropiadas.
- sistemas de estimación de modelos probabilísticos: esta aproximación se basa en la asunción de que el problema de decisión puede expresarse en términos probabilísticos. En estos casos hay que estimar dos tipos de distribuciones de probabilidad, la probabilidad a priori de que una muestra pertenezca a una clase determinada, y la probabilidad condicional. Este tipo de modelos se pueden estimar mediante la técnica *Maximum Likelihood* (ML) o mediante estimación Bayesiana [DH74, F’u99, TK03].

En el caso de que el sistema sea un reconocedor, o lo que es lo mismo, realiza una interpretación (las probabilidades/clasificación de las unidades básicas se obtienen utilizando métodos de clasificación), los sistemas de estimación de modelos se clasifican del siguiente modo:

- sistemas de estimación de modelos para reconocedores sintácticos: estos sistemas entrenan modelos de palabras que expliquen como se forman las palabras a partir de caracteres, y modelos de como se combinan estas para formar frases, los llamados modelos de lenguaje. La mayoría de sistemas utilizan modelos de palabras donde cada modelo consiste en la sucesión de los caracteres que la conforman. Los modelos de lenguaje se estiman mediante inferencia gramatical u otras técnicas estadísticas, entre las que destacan lo n-gramas [Jel98, Kat87, MB01, NEK94, SB93].

- sistemas de estimación de modelos relacionales: estos modelos son muy usados por los sistemas de RATM donde el texto esta escrito en alguna lengua asiática como el chino o el japonés, por ejemplo. En estas lenguas, las palabras, se suelen representar mediante grafos jerárquicos [KK98, LRS91, NT96].

Con respecto a la partición del corpus para entrenamiento podemos clasificar los sistemas de RATM como:

- dependiente del escritor: el sistema de RATM se entrena con muestras de un único escritor, de manera que el sistema se especializa en el estilo de dicho escritor. Este tipo de sistemas proporcionan tasas de acierto muy elevadas (para el escritor con el que se ha entrenado). El principal problema que presentan estos sistemas es la dificultad de conseguir suficientes muestras para que el sistema resulte productivo.

- independiente del escritor: el sistema se entrena con muestras de diversos escritores de manera que, sacrificando un poco de precisión, pueda reconocer distintos estilos de escritura. Al recolectar muestras de distintos escritores, es relativamente fácil recoger suficientes muestras para entrenar el sistema.

- sistemas adaptables: partiendo de un sistema independiente del escritor, éste se adapta, a partir de unas pocas muestras, para mejorar el reconocimiento para un escritor determinado. Estos sistemas se pueden aprovechar del aprendizaje continuo o incremental, para ir adaptando mejor los modelos, según se vaya utilizando el sistema.

La utilización de léxicos es una de las restricciones más usuales impuestas a los sistemas RATM. El tamaño del léxico limita el espacio búsqueda proponiendo una serie de posibles palabras candidatas a la hipótesis de reconocimiento. Con respecto al tipo de léxico los sistemas RATM se clasifican del siguiente modo:

- vocabulario restringido: el sistema RATM dispone de una lista de palabras permitidas. Es la restricción más usual en la mayoría de sistemas RATM.

- vocabulario abierto: el sistema de RATM no se basa sobre un léxico (o está basado en uno relativamente grande) [BSM98, BRKR00].

La talla del vocabulario, es uno de los principales referentes para determinar el grado de dificultad de una tarea. Este tamaño determina el tiempo de respuesta del sistema RATM y la precisión del mismo. Al aumentar la talla del léxico, se aumenta el número de palabras parecidas, con lo que aumenta la posibilidad de confusión del sistema. Además, el espacio de búsqueda se incrementa, con lo que el coste computacional también se ve incrementado. Es usual encontrar en la literatura los siguientes grados de dificultad de las tareas dependiendo de la talla del léxico [KSS03a]:

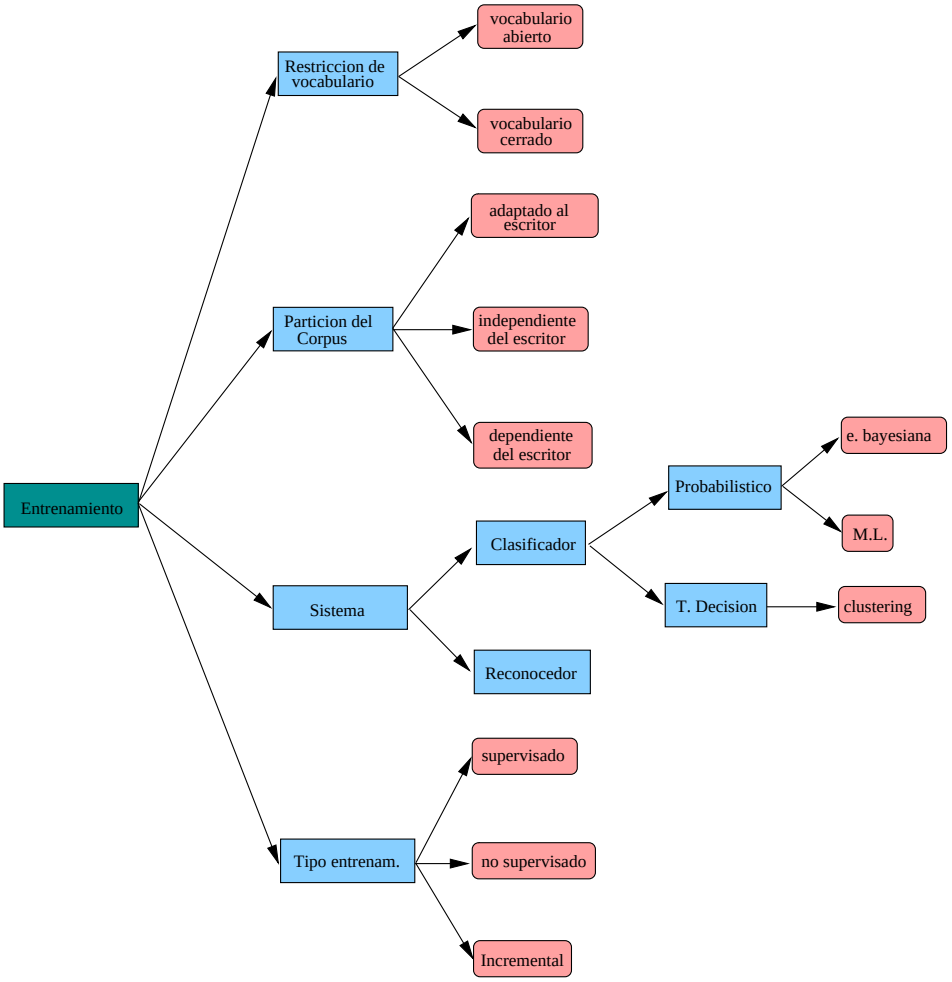


Figura 1.2: Esquema de la taxonomía de los sistemas RATM desde el punto de vista del entrenamiento

- vocabulario pequeño: unas pocas decenas de palabras.
- vocabulario medio: centenares de palabras.
- vocabulario grande: miles de palabras.
- vocabulario muy grande: decenas de miles de palabras.

1.2. Estado del arte

En esta sección se resume el estado del arte de los sistemas de RATM[1]. Aunque el OCR cuenta ya con una cierta antigüedad, el RATM es una disciplina relativa-

[1]Los capítulos donde se tratan temas especificos incluyen una revisón del estado del arte.

mente reciente. De hecho, los mayores avances en el campo se han producido en la última década [SRI99, PS00].

Hoy por hoy, los humanos consiguen mejores tasas de reconocimiento de texto manuscrito que los mejores sistemas automáticos, funcionando con los más potentes computadores. En cualquier caso, los sistemas de RATM actuales funcionan de manera razonable bajo una serie de restricciones (bastante fuertes en general) [KSS03b]. Estas condiciones son básicamente cuatro:

a) dominios semánticos muy delimitados.

b) vocabularios restringidos, de tamaño pequeño o medio.

c) un único estilo de escritura (cursiva o impresa).

d) dependencia del escritor.

Los sistemas RATM actuales son capaces de reconocer palabras que se encuentren entre un número limitado de palabras, típicamente unos pocos miles. Los léxicos suelen estar integrados dentro del proceso de reconocimiento de manera que se rechacen, lo más temprano posible, las hipótesis con poca probabilidad. Algunos de estos sistemas se pueden encontrar como productos comerciales funcionando en aplicaciones reales sobre determinadas tareas (para más detalles ver sección 1.3).

Un gran número de algoritmos usados por los los sistemas RATM, están basados, parcialmente o en su totalidad, en heurísticos más o menos sofisticados, lo que aumenta el riesgo de dependencia con los datos de la tarea. También cabe decir que los sistemas RATM requieren un gran esfuerzo experimental para encontrar el conjunto de parámetros óptimo cada vez que se usan en una tarea nueva. Para conseguir que los sistemas RATM sean mucho más automáticos y generales se requerirá un gran esfuerzo en investigación en los próximos años.

Los nuevos sitemas RATM que se están desarrollando siguen estando basados en vocabularios cerrados, pero de tamaño considerablemente grande (miles, o decenas de miles de palabras), y sin restricciones en cuanto al estilo de escritura.

El problema del RATM general sin restricciones parece que no será resuelto a corto plazo. De todas maneras, hay tres grandes frentes abiertos dependiendo de la unidad básica a reconocer. El primero toma como unidad básica el caracter, el segundo la palabra, y el tercero la frase. A continuación se exponen con más detalle cada una de estas aproximaciones.

1.2.1. Clasificación de caracteres manuscritos aislados

La unidad más natural del texto manuscrito es el caracter. En esta aproximación el problema del RATM se intenta resolver mediante la utilización de técnicas analíticas. Estas técnicas han demostrado su eficacia en el reconocimiento de texto impreso.

En las dos últimas décadas aparecieron corpus de caracteres manuscritos de uso libre, como el NIST [Gar92, GJ92] o CEDAR [HDM+94], los cuales permitieron un gran avance en el campo del OCR aplicado a la escritura manual. Las técnicas de OCR cuentan con una cierta antigüedad dentro de la disciplina del reconocimiento de texto. En los trabajos de [IOO91, MSY92], se presentan revisiones históricas del OCR. En [Nag00] se presenta un revisión del estado actual, y de las técnicas utilizadas hasta el momento. Trabajos más recientes en este dominio, se pueden encontrar en [BC03, LNSF02, US02, PP02, PP03, US03]. Comparativas para el corpus NIST en los trabajos de [eA94, WGJ+92]. Comparativas para CEDAR en [GWJ+94]

La clasificación de caracteres previamente segmentados es muy problemática. El problema surge de la necesidad de segmentar las palabras en caracteres, que a diferencia de lo que ocurre en texto impreso, es un problema muy difícil. Casey et al. en su trabajo [CL96] y Lu et al. en [LS96] presentan un abanico de técnicas de segmentación de texto en caracteres. La mayoría de técnicas son heurísticas. En 1973 Sayre publica un artículo [Say73] en el que aparece su famosa paradoja: ”una palabra no puede ser segmentada sin antes haber sido reconocida, y no puede ser reconocida sin antes haber sido segmentada”, que parece que sentencia esta aproximación. Aun así, esta aproximación cuenta con un amplio abanico de algoritmos consolidados [CKS95, BRST95, EYGSS99] con lo que el problema se reduce a encontrar métodos que proporcionen un conjunto de segmentaciones aceptables, y utilizar información de mayor nivel (léxica, sintáctica, etc) para decidir sobre la mejor de todas [Bun03].

1.2.2. Reconocimiento de palabras aisladas

El reconocimiento de palabras aisladas se encuentra con el inconveniente de la poca disponibilidad de corpus públicos, que permitan comparar las diferentes técnicas desarrolladas. El corpus CEDAR es prácticamente el único corpus público disponible, aunque recientemente ha aparecido una versión segmentada a nivel de palabra de IAMDB [ZB00].

En este campo se siguen dos grandes aproximaciones, una analítica, donde cada palabra está compuesta por unidades más pequeñas (los caracteres), que hay que segmentar previamente, para luego ser clasificadas por separado, y otra holística, donde cada palabra es vista como una unidad. En [MG96] se puede encontrar un estudio comparativo de las dos aproximaciones.

El estudio realizado en reconocimiento de palabras aisladas siguiendo una aproximación analítica, muestra el mismo problema que el del reconocimiento de caracteres aislados, puesto que ha de dividir la palabra en caracteres, para luego clasificarlos por separado. Así, la mayoría de trabajos se centran en los algoritmos de segmentación de palabras en caracteres [BS89, MG96, YS98], puesto que el problema de la clasificación de caracteres manuscritos aislados, se puede considerar como resuelto. En [KFK02], se puede encontrar trabajo reciente, donde se pretende reconocer cualquier posible palabra sin restricción, y de los mismos auto-

res, [KSFK03] donde se realiza un postproceso con la ayuda del léxico.

En cuanto a la aproximación holística se puede decir que es muy robusta respecto al estilo de escritura, como al instrumento utilizado para escribir, o frente a cualquier tipo de ruido o perturbación. Estos sistemas funcionan muy bien para tareas donde el vocabulario es muy pequeño, del orden de unas pocas decenas de palabras, pero si la talla del léxico se incrementa, su productividad cae rápidamente. Algunos de estos sistemas están basados en modelos de Markov, con un modelo por cada palabra [GB03, MSBS03, Sch03a, Sch03b]. En [Mor91] y en [SLB91] se presentan trabajos que siguen una aproximación geométrica, donde cada palabra se convierte en una cadena, la cual es comparada con las palabras del léxico, mediante distancia de edición o de Levehstein.

1.2.3. Reconocimiento general de texto manuscrito

El reconocimiento general de texto manuscrito puede ser considerado un tema relativamente nuevo. Se puede encontrar una visión general del estado del arte en los trabajos [KGS99, PS00, SRI99, Vin02]. En [Bun03] se puede encontrar una revisión completa de lo que se ha hecho, lo que se se está haciendo, y una previsión de como evolucionará el campo en un futuro cercano.

La aproximación clásica se basa en la mayoría de los casos en soluciones analíticas, donde cada linea de texto se segmenta en palabras, las cuales son reconocidas utilizando un sistema de reconocimiento de palabras aisladas [BS89, KFK02, SR98].

Los sistemas que destacan en la actualidad suelen estar basados en algoritmos sin segmentación explícita. Un gran número de ellos están basados en modelos de Markov [BSM98, BRST95, GS97, MSLB98, MB01, MG96, EYGSS99, VBB03]. En esta tesis se seguirá esta linea.

1.3. Sistemas comerciales

Los sistemas de RATM tienen éxito en el caso de aplicaciones en dominios semánticos muy restringidos, o que requieran vocabularios muy pequeños, del orden de 30 ó 40 palabras; para un solo estilo de escritura, o para sistemas dependientes del escritor.

La mayoría de sistemas comerciales presentan vocabularios cerrados, y un dominio semántico restringido donde se tiene información redundante e/o información del contexto de la aplicación. Se ha hecho mucho trabajo dedicado a automatizar el tratamiento de los cheques bancarios, y a la clasificación automática del correo.

1.3.1. Reconocimiento de cantidades numéricas en cheques

El reconocimiento de la cantidad legal de los cheques es un problema relativamente sencillo, a pesar de que cada muestra está escrita por un escritor distinto,

debido a lo reducido de su vocabulario, a lo regular de su lenguaje, y a la redundancia introducida por la cantidad de cortesía (el mismo valor escrito en dígitos). Esto justifica la calidad de los productos comerciales, que desde muy temprano, existen en el mercado. Algunos de los productos desarrollados son de gran calidad, como los descritos en [GAA+99, GAA+01], con tasas de acierto cercanas a las de los humanos. En estos trabajos se puede encontrar toda una familia de sistemas comerciales multilingües de lectura de cheques. Trabajos recientes se pueden encontrar en [LDG+00, GS95, KB00, KABP98, SLG+96].

1.3.2. Reconocimiento de direcciones postales

El reconocimiento de direcciones postales presenta como principales características que el tipo de escritura además de ser completamente espontáneo como en el caso anterior, se tienen muy pocas muestras por escritor, usualmente una, y con respecto al tipo de letra, este puede ser cursivo, impreso o incluso mixto. También cabe decir que cada muestra es de un escritor distinto. El vocabulario está particionado por distritos postales, y se elige dependiendo del código postal [Sri00]. Destacables son los trabajos basados en modelos de Markov [CKZ94, CHS94, KG97, Kor97, MG96, EYGSS99]. Se pueden encontrar tablas comparativas para distintos sistemas comerciales en [aEKL00].

1.4. Objetivos de la tesis

El problema del *reconocimiento automático del habla continua* (RAH) y el del *reconocimiento automático de texto manuscrito* (RATM) presentan grandes similitudes. Debido a ello, se está realizando un gran esfuerzo para adaptar la tecnología utilizada en RAH al dominio de los sistemas RATM. Este trabajo se centrará en esa misma linea de investigación, y se abordará el problema del RATM usando tecnología basada en los sistemas libres de segmentación, explotados con éxito en el campo del RAH [GST+00, TJG+04, TJV04, TPJV05]. Con este fin, se adaptará el reconocedor del habla continua ATROS (*Automaticaly Trainable Recognizer Of Speech*) [PSCV01] para ser utilizado como reconocedor de texto manuscrito.

Los sistemas de reconocimiento automático de texto manuscrito estarán maduros para su uso generalizado, cuando sean fáciles de usar, esto es, cuando éstos sean capaces de ofrecer a cualquier usuario, sin ningún tipo de preparación o adiestramiento para su utilización, una productividad razonable. Estos sistemas han de ser flexibles y robustos en cuanto a la entrada, de tal manera que no se requiera del escritor ningún esfuerzo extra, que no haría si escribiese para ser leído por un humano.

En esta tesis se explorarán diferentes métodos de normalización de la señal de entrada, con intención de dotar al al sistema de mayor robustez, y conseguir que su utilización pueda ser más relajada.

Los sistemas dependientes del escritor consiguen mejores tasas de acierto que

los independientes del escritor. Por otra parte, los sistemas independientes del escritor tienen más facilidad para reunir muestras de entrenamiento. En esta tesis, se estudiará la adaptación de sistemas independientes del escritor para su utilización por un único escritor, con la intención de que a partir de una pocas muestras producidas por este escritor se mejore la productividad del sistemas para este escritor, o lo que es lo mismo, que éste pueda escribir de manera más relajada, sin que el sistema pierda productividad.

Los sistemas de reconocimiento de texto manuscrito no están exentos de errores. No sólo interesa saber el número de errores que producirá el sistema, sino que es importante saber qué unidades de la hipótesis producida están mal reconocidas, o no se tiene garantía de que estén bien reconocidas, para así poderlas corregir manualmente. Esta es una condición básica para poder construir sistemas de RATM que puedan ser utilizados por cualquier persona. Esta tecnología permitirá construir, por ejemplo, asistentes a la transcripción, que realizarán el grueso del trabajo, dejando para el operador humano la supervisión y corrección de aquellas palabras de las cuales se tenga poca garantía de que estén bien reconocidas. En esta tesis se estudiará la adaptación de las técnicas de verificación de hipótesis más usadas en el campo del RAH para ser usadas por los sistemas RATM.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

En este capítulo se revisan las bases teóricas, formulaciones y algoritmos principales relacionados con los *modelos Ocultos de Marcov* y con los modelos de lenguaje n-gramas. Estos modelos constituyen la base de los sistemas de reconocimiento automático de texto manuscrito desarrollados, y usados a lo largo de este trabajo. En la sección 2.3 se exponen las distintas métricas utilizadas para evaluar los sistemas RATM. La métrica utilizada para evaluar los sitemas de verificación automática de hipótesis (capítulo 7), por ser especifica para este tipo de sistemas, se expondrá en dicho capítulo.

El problema del reconocimiento automático de texto manuscrito puede formularse en el marco estadístico como la búsqueda de la secuencia de palabras más probable segun:

$$\hat{W} = \underset{W}{\text{argmax}}\, P(W|X) \tag{2.1}$$

donde X es una secuencia de vectores de características que representan la imagen que contiene el texto manuscrito y $W = \{w_1, w_2, \cdots, w_n\}$ una secuencia de palabras. $P(W|X)$ es la probabilidad que teniendo la secuencia de vectores X, estos correspondan a la secuencia de palabras W. Mediante la regla de Bayes la probabilidad $P(W|X)$ se puede reescribir como:

$$P(W|X) = \frac{P(X|W)\, P(W)}{P(X)} \tag{2.2}$$

donde $P(W)$ es la probabilidad de la secuencia de palabras, $P(X|W)$ es la probabilidad de observar la secuencia de vectores de características X dada la secuencia de palabras W. Mientras que $P(X)$ es la probabilidad de la secuencia de vectores de características. La ecuación 2.1 quedaría como:

$$\hat{W} = \underset{W}{\text{argmax}}\, \frac{P(X|W)\, P(W)}{P(X)} \tag{2.3}$$

En la ecuación 2.3 se observa que $P(X)$ es una constante que no influye en la maximización y puede ser eliminada. De tal manera que la ecuación fundamental

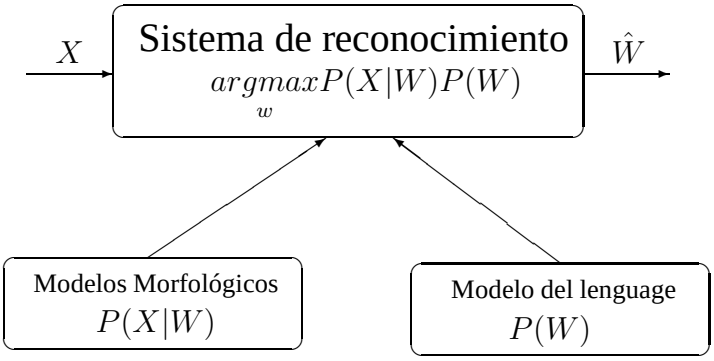


Figura 2.1: Esquema general de un sistema de reconocimiento de texto manuscrito

de los sistemas de RATM es:

$$\hat{W} = \underset{W}{\text{argmax}} P(X|W)\ P(W) \tag{2.4}$$

En este trabajo la probabilidad $P(X|W)$ se modelará con modelos ocultos de Markov, mientras que la probabilidad de la secuencia de palabras, $P(W)$ se modelará mediante los modelos del lenguaje denominados n-gramas (ver figura 2.1).

2.1. Modelos ocultos de Markov

Un **modelo de Markov de capa oculta** o en inglés *Hidden Markov Model* (HMM) [HAJ90, Jel98, Lee89] es una herramienta para representar distribuciones de probabilidad sobre secuencias de observaciones. Las observaciones pueden ser discretas, valores reales, enteros o cualquier objeto sobre el cual se pueda definir una distribución de probabilidad [Gha01]. Se asume que las observaciones son muestreadas a intervalos regulares de tal manera que el índice tiempo es un valor entero.

Un Modelo de Markov de capa oculta es un autómata de estados finitos en el cual concurren dos procesos estocásticos. Uno de estos procesos no puede ser observado (de aquí viene el nombre de capa oculta), mientras que el otro proceso produce una secuencia de observaciones de salida. En este último proceso, asociado a cada estado del autómata, se puede emitir una observación de un conjunto de observaciones de salida siguiendo una cierta función de probabilidad. Se asume que cada estado satisface la **propiedad de Markov** según la cual, para cualquier secuencia de eventos ordenados en el tiempo, la densidad de probabilidad condicional de un evento dado, depende solamente de los i eventos más cercanos. Un proceso que satisfaga esta propiedad es un *proceso de Markov*. El orden del proceso de Markov viene definido por i. Así hablamos de proceso de Markov de i-orden.

Por motivos de tractabilidad, los modelos de Markov más extendidos son los de primer orden. A partir de ahora siempre se hablará de HMMs de primer orden. En los HMMs se asumen dos suposiciones. La primera es que la probabilidad de cada observación es independiente de las observaciones anteriores (primer orden markoviano). La segunda asunción es la independencia de la salida, la probabilidad de salida para un símbolo sólo depende del estado emisor y no se tiene en cuenta como se ha llegado a él [HAJ90].

Dada una secuencia de observaciones $O = \{o_1, o_2, ..., o_n\}$ no es trivial encontrar la secuencia de estados que produjo dicha salida, ya que cada estado puede emitir cualquier símbolo de salida. Lo cual significa que puede haber más de una secuencia de estados que podrían haber emitido la misma secuencia de observaciones.

Los HMMs, a pesar de la restricción de la no existencia de correlación entre los distintos símbolos de la cadena modelada, son ampliamente utilizados en un gran número de sistemas de reconocimiento automático de patrones. En un principio, los HMMs se aplican con éxito al reconocimiento automático del habla. Debido a la similitud entre el reconocimiento automático del habla y del texto manuscrito, los HMMs se han hecho también muy populares en el campo del reconocimiento automático de texto manuscrito. Los sistemas que se han desarrollado a lo largo de este trabajo están basados en la tecnología de HMMs, por lo cual en esta sección se expondrá la teoría, formulación y algoritmos básicos de los HMMs.

Dependiendo de la naturaleza de las observaciones modeladas, los HMMs se clasifican en varios tipos: si las características toman valores discretos hablamos de HMMs discretos, si por el contrario toman valores continuos, hablamos de HMMs continuos. En el primer caso, las observaciones corresponden a un alfabeto determinado o a la discretización de observaciones continuas (codebooks, patrones obtenidos a partir de una cuantificación vectorial). En el caso de HMMs continuos, la ley probabilística que gobierna la emisión de observaciones por parte de los estados es una función de densidad de probabilidad, normalmente aproximada por una mixtura de gausianas. Existe un tercer tipo de HMMs, los llamados semicontinuos, los cuales utilizan observaciones discretas, pero son modeladas mediante una función de densidad de probabilidad. En este caso todos los estados del modelo comparten la misma función de densidad.

La formulación asociada a los HMMs en general viene expuesta en [HAJ90, Jel98, Lee89] no obstante como a lo largo de este trabajo se van a utilizar los HMMs continuos, se darán las definiciones y algoritmos más usuales para estos últimos.

2.1.1. Definición de HMM continuo

Formalmente un HMM continuo $\mathcal{M}$ es una máquina de estados finitos definida por la séxtupla (Q, I, F, X, a, b), donde:

- Q es un conjunto finito de estados, que incluye un conjunto de estados iniciales $I \subseteq Q$ y un conjunto de estados finales $F \subseteq Q$.

- X es un espacio real d-dimensional de observaciones: $X \subseteq \mathbb{R}^d$.
- $a : Q \times Q \rightarrow [0, 1]$ es una función de distribución de probabilidad de transición entre estados, tal que:

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

- $b : Q \times X \rightarrow [0, 1]$ es una función de densidad de probabilidad de emitir un vector $\vec{x} \in X$ en un estado $q_i \in Q$, tal que:

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

En la definición de HMM dada, hay implícitos dos supuestos:

1. $a(q_i, q_j) = P(s_{t+1} = q_j | s_t = q_i)$ establece[1] que la probabilidad de una cadena de Markov en un particular estado q_j en $t+1$ depende sólo del estado q_i de la cadena de Markov en el tiempo t, y no depende de los estados visitados previamente en tiempos anteriores a t, es decir:

$$P(s_{t+1} | s_1 \dots s_t) = P(s_{t+1} | s_t)$$

2. $b(q_i, \vec{x}) = p(x_t = \vec{x} | s_t = q_i)$ establece[2] que la probabilidad de que $\vec{x}$ sea emitida en el tiempo t depende sólo del estado q_i en el tiempo t, y no depende ni de los vectores emitidos y ni de los estados visitados previamente en tiempos anteriores a t, es decir:

$$p(x_t | x_1 \dots x_t, s_1 \dots s_t) = p(x_t | s_t)$$

Para simplificar la notación, $a(q_i, q_j)$ se reescribirá como $a_{i,j}$, $b(q_i, o_t)$ como $b_i(o_t)$, mientras que la probabilidad de que el estado i sea inicial se expresará como $a_{0,i}$ y de que sea final como a_i

2.1.2. Algoritmos básicos para HMMs

Nos encontramos con tres problemas bien definidos: El problema de la evaluación, el problema de la estimación y el problema de la decodificación.

a) Evaluación: Aquí nos enfrentamos al problema de determinar $P(O|\lambda)$, dada una secuencia de observaciones $O = \{o_1, o_2, o_3 ... o_T\}$ y el modelo λ, esto es, la probabilidad de que esa secuencia haya sido producida por dicho modelo.

[1] $s_t = q_i$ denota que el HMM se encuentra en el estado q_i en el tiempo t.
[2] $x_t = \vec{x}$ denota que el HMM en el estado s_t emite $\vec{x}$ en el tiempo t.

b) Estimación: Dada una secuencia de observaciones O, como ajustar los parámetros del modelo λ para que maximice la probabilidad de generar la secuencia $P(O|\lambda)$.

c) Decodificación: Dada una secuencia de observaciones O, cual es la secuencia de estados del modelo que con mayor probabilidad la han producido. Se trata de averiguar la parte oculta de los HMMs.

a) Evaluación: La manera más directa que hay de calcular la probabilidad de una observación, $O = \{o_1, o_2, ..., o_T\}$ dado un modelo λ, es explorar todas las posibles secuencias de estados $S = \{s_1, s_2, ..., s_T\}$ de longitud T, calcular la probabilidad de que cada secuencia de estados haya generado la muestra $P(O|S, \lambda)$, y sumarlas.

La probabilidad de cada secuencia de estados viene dada por

$$P(S|\lambda) = a_{s_0,s_1}\ a_{s_1,s_2}\ a_{s_2,s_3}\ ...\ a_{s_{t-1},s_T} \tag{2.5}$$

mientras que la probabilidad de emisión dada una secuencia de estados S y un modelo λ para la secuencia O es

$$P(O|S, \lambda) = b_1(o_1)\ b_2(o_2)\ b_3(o_3)\ ...\ b_T(o_T) \tag{2.6}$$

La probabilidad conjunta de que dado un modelo se produzca una secuencia de estados que produzcan la observación no es más que el producto de las dos probabilidades anteriores.

$$P(O, S|\lambda) = P(O|S, \lambda)P(S|\lambda) \tag{2.7}$$

Así, la probabilidad de que un modelo haya generado una secuencia de observaciones es:

$$P(O|\lambda) = \sum_{\forall S} P(O|S, \lambda)P(S|\lambda) = \sum_{\forall S} \left[a_{s_0,s_1} b_1(o_1) \prod_{t=2}^{T} a_{s_{t-1},t} b_t(o_t) \right] \tag{2.8}$$

El algoritmo **Forward** es un algoritmo eficiente para calcular $P(O|\lambda)$. Primero hay que definir la función recursiva *forward*:

$$\alpha_j(t) = P((O_1^T, s_t = q_j|\lambda) \tag{2.9}$$

$$\alpha_j(t) = \begin{cases} a_{0,j}\ b_j(o_1) & t = 1 \\ \left(\sum_i \alpha_i(t-1)\ a_{i,j} \right)\ b_j(o_t) & 1 < t \leq T \end{cases}$$

La función $\alpha_j(t)$ es la probabilidad de que un proceso de Markov, estando en el estado s_j, en el tiempo t, haya emitido la secuencia de observaciones, $o_1,\ o_2,\ ...,\ o_t$. La probabilidad de que una secuencia O sea emitida por un modelo λ es:

$$P(O|\lambda) = \sum_{\{i:\ q_i \in F\}} a_i \alpha_i(T) \tag{2.10}$$

El algoritmo **backward** es complementario del *forward*. La función $\beta_i(t)$ denota la probabilidad de que un proceso de Markov, estando en el estado s_i, en el tiempo t, vaya a emitir la secuencia de observaciones $o_{t+1}, o_{t+2} \cdots o_T$

$$\begin{aligned} \beta_i(t) &= P(O_{t+1}^T | s_t = q_i, \lambda) \\ \beta_i(t) &= \begin{cases} a_j & t = T \\ \sum_j a_{i,j}\ b_j(o_{t+1})\ \beta_j(t+1) & 1 < t \leq T \end{cases} \end{aligned} \tag{2.11}$$

Utilizando la variable β, la probabilidad de que una secuencia O sea emitida por un modelo λ es:

$$P(O|\lambda) = \sum_{\{i:\ q_i \in I\}} a_{0,i} b_i(o_1) \beta_i(1) \tag{2.12}$$

b) Estimación: El problema más difícil es como ajustar los parámetros que definen un modelo HMM (transiciones entre estados y las distribuciones de probabilidad de emisión de símbolos de cada estado) de tal manera que se maximice la probabilidad de que una observación sea generada por el modelo. El algoritmo iterativo **Backward-Forward** o **Baum-Welch** resuelve este problema mediante un proceso de maximización de la esperanza (EM, del inglés *Expectation Maximization*).

Dado un modelo λ cuyos parámetros no son nulos (modelo inicializado), la probabilidad *a posteriori* de una transición $\gamma(i,j)$, condicionada a una secuencia de observaciones O, puede ser calculada como:

$$\begin{aligned} \gamma_{i,j}(t) =& P(s_t = i, s_{t+1} = j | O, \lambda) \\ =& \frac{\alpha_i(t) a_{i,j} b_j(o_{t+1}) \beta_j(t+1)}{P(O|\lambda)} \\ =& \frac{\alpha_i(t) a_{i,j} b_j(o_{t+1}) \beta_j(t+1)}{\sum_{k \in S_F} a_k \alpha_k(T)} \end{aligned} \tag{2.13}$$

De tal manera que $\gamma_{i,j}(t)$ es la probabilidad de estar en el estado i en el tiempo t y pasar al estado j en el momento $t+1$, dados el modelo y la secuencia de estados. Esta formula tiene en cuenta la probabilidad de todos los caminos que van a parar al estado i en el momento t, o lo que es lo mismo, la probabilidad *forward* de estar en i en el momento t, multiplicado por la probabilidad a priori de ir al estado j, esto es $a_{i,j}$, consumiendo el símbolo o_{t+1}, esto es $b_j(o_{t+1})$, por la probabilidad de estar en el estado j en el momento $t+1$ o lo que es lo mismo, la probabilidad *backward* $\beta_j(t+1)$.

De la misma manera, la probabilidad *a posteriori* de estar en un estado i en un tiempo determinado t, dado una secuencia de observaciones y un modelo, puede ser calculado como:

$$\begin{aligned}\gamma_i(t) =& P(s_t = i|O,\lambda) \\ =& \frac{\alpha_i(t)\beta_i(t)}{\sum\limits_{k\in S_F} a_k\alpha_k(T)}\end{aligned} \tag{2.14}$$

De la ecuación 2.14 se puede observar que la probabilidad de estar en un estado se puede calcular como:

$$\gamma_i(t) = \sum_j \gamma_{i,j}(t) \qquad si \quad t < T \tag{2.15}$$

En la ecuación 2.15 se reutiliza el cálculo de $\gamma_{i,j}(t)$ y sólo es necesario hacer sumas, cosa que es computacionalmente mucho mas barato que calcular las probabilidades *forward* y *backward* directamente.

Así pues la probabilidad de transitar de i a j en el momento t puede ser calculado como todas las veces que se ha activado esa transición, $\gamma_{i,j}(t)$, dividido por el total de transiciones que se han activado partiendo del estado i o lo que es lo mismo, dividido por el número de veces que se ha estado en el estado i, esto es, $\gamma_i(t)$.

$$\overline{a_{i,j}} = \frac{\sum\limits_{t=1}^{T-1}\gamma_{i,j}(t)}{\sum\limits_{t=1}^{T-1}\gamma_i(t)} = \frac{\sum\limits_{t=1}^{T-1}\gamma_{i,j}(t)}{\sum\limits_{t=1}^{T-1}\sum\limits_{j}\gamma_{i,j}(t)} \tag{2.16}$$

En cuanto a la probabilidad de emisión de un símbolo en un estado $b_i(o_k)$ se puede calcular como la frecuencia del símbolo k en el estado i con respecto a la frecuencia de todos los símbolos que se hayan dado en dicho estado. La suma a lo largo del tiempo de $\gamma_i(t)$ es el número de veces que el estado i ha sido visitado, lo cual es equivalente al número total de símbolos emitidos.

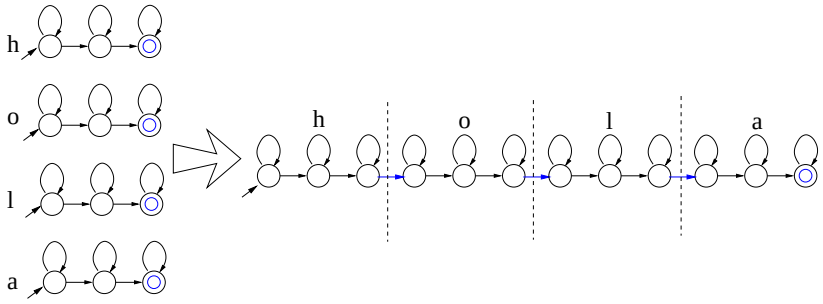


Figura 2.2: Ejemplo de construcción de un macromodelo HMM.

El problema de estimar la probabilidad de emisión $b_i(o_k)$ en el caso de que se modele con una única gausiana puede ser expresado como:

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(o_t-\mu_j)'\Sigma_j^{-1}(o_t-\mu_j)} \tag{2.17}$$

si el modelo tuviese solamente un estado el problema se reducicía a estimar las medias y varianzas del siguiente modo:

$$\hat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} o_t \qquad \hat{\Sigma}_j = \frac{1}{T}\sum_{t=1}^{T}(o_t-\mu_j)(o_t-\mu_j)' \tag{2.18}$$

En el caso de que la probabilidad de emisión se modele por una mixtura de gausianas, se puede considerar como una forma especial de subestados donde los pesos de las gausianas dentro de la mixtura se toman como las probabilidades de transición.

La estimación explicada aquí es sólo para un HMM aislado. En el caso de la escritura manuscrita en la que se pretenda modelar cada caracter, pero no se disponga de una segmentación a nivel de caracteres, sino que se disponga de palabras o de frases, se procede de la siguiente manera: se conforma un modelo con la unión de los modelos que conforman la palabra o la frase, siguiendo el mismo orden en el que se leen. Los estados iniciales del primer modelo (el que está más a la izquierda) pasan a ser los estados iniciales del macromodelo. Los estados finales del último modelo (el que está más a la derecha) pasan a ser los estados finales del macromodelo. Las transiciones entre modelos se resuelven conectando los estados finales del modelo de la izquierda con los estados iniciales del modelo siguiente (ver ejemplo de la figura 2.2). Hay que decir que la mayoría de *software* para estimar HMM fuerzan a que los HMM tengan un único estado inicial y un único estado

final, por motivos de simplicidad. Al final de este proceso, se divide el macromodelo y se repite la operación hasta que se hayan procesado todas las palabras y/o frases.

c) Decodificación: Dado el modelo λ, se pretende obtener la secuencia de estados óptima, S, asociada a una secuencia de observaciones, O. Es decir, encontrar un camino S, que maximize $P(O, S|\lambda)$. El algoritmo usado para decodificación en este contexto, es el algoritmo de **Viterbi**. Este es un algoritmo similar al de *forward* salvo que en la función *forward* se sumaban todos los caminos que llegaban a un estado en un tiempo determinado, mientras que ahora se maximiza.

$$\psi_j(t) = \left\{ \begin{array}{ll} a_{0,j}\ b_j(o_1) & t = 1 \\ \underset{i}{max} \left(\psi_i(t-1)\ a_{i,j} \right)\ b_j(o_t) & 1 < t \leq T \end{array} \right\} \tag{2.19}$$

La probabilidad de viterbi de que una secuencia O sea emitida por un modelo λ es:

$$P(O|\lambda) = \underset{\{i:\ q_i \in F\}}{max} \psi_i(T) \tag{2.20}$$

En la mayoría de los sistemas, la topología de los HMMs (el número de estados y las transiciones), así como el número de gausianas se elige empíricamente. Sin embargo, algunos trabajos [GB04, San98, Tak99, ZB02] exploran la optimización de estos parámetros.

2.2. Modelos de lenguaje

Tener conocimiento de la estructura del discurso y de qué es lo más probable que aparezca en un contexto determinado es de gran valor en la fase de decodificación, para ayudar en la búsqueda de la mejor secuencia de símbolos $W = w_1...w_m$ que corresponden a una secuencia de vectores de características. Los modelos de lenguaje se usan como restricciones para reducir el espacio de búsqueda. De hecho, los modelos de lenguaje se encargan de asignar una probabilidad a toda posible secuencia de palabras.

La probabilidad de observar una secuencia de símbolos $W = w_1...w_m$ de un vocabulario conocido Σ en un lenguaje determinado puede expresarse como:

$$P(W) = P(w_1) \cdot \prod_{i=2}^{m} P(w_i|w_1...w_{i-1}) \tag{2.21}$$

donde $P(w_i|w_1...w_{i-1})$ es la probabilidad que habiendo visto la secuencia de palabras $w_1...w_{i-1}$ aparezca a continuación la palabra w_i. A la secuencia de palabras

previa a w_i se le suele llamar historia. En la práctica es imposible estimar correctamente $P(w_i|w_1...w_{i-1})$ ya que muchas de las historias, incluso para una i pequeña, aparecen con una frecuencia muy baja, o incluso no aparecen. Hay que tener en cuenta que para un vocabulario de talla $|\Sigma|$ existen $|\Sigma|^{i-1}$ posibles historias. La estimación de $P(W)$ se hace pues impracticable. No obstante hay una aproximación que pese a su simplicidad, en la práctica funciona sorprendentemente bien: los modelos de lenguaje n-gramas.

2.2.1. n-gramas

Estos modelos definen una función $\Phi_n : \Sigma^* \rightarrow \Sigma^{n-1}$ que clasifica en una misma clase de equivalencia todas aquellas cadenas que terminan con las mismas $n-1$ palabras. Ahora la probabilidad de $P(W)$ puede aproximarse como:

$$P(W) \approx \prod_{i=1}^{m} P(w_i|\Phi_n(w_1...w_{i-1})) = \prod_{i=1}^{m} P(w_i|w_{i-n+1}\cdots w_{i-1}) \tag{2.22}$$

Debido a que en las primeras palabras de la cadena sucede que $i - n \leq 0$, la expresión 2.22 se reescribe como:

$$P(W) \approx P(w_1) \cdot \prod_{i=2}^{n-1} P(w_i|w_1\cdots w_{i-1}) \cdot \prod_{i=n}^{m} P(w_i|w_{i-n+1}\cdots w_{i-1}) \tag{2.23}$$

En la práctica se suelen utilizar valores de n reducidos (1,2 ó 3) debido al problema endémico de la carencia de muestras disponibles de las que estimar el modelo. La estimación de la probabilidad de que se dé una palabra w_i habiendo ocurrido una historia determinada $w_{i-n+1}\cdots w_{i-1}$, se calcula mediante la frecuencia relativa $f(|)$ de ocurrencia del n-grama $w_{i-n+1}\cdots w_i$ normalizada por las veces que ha ocurrido su historia:

$$P(w_i|w_{i-n+1}\cdots w_{i-1}) = f(w_i|w_{i-n+1}\cdots w_{i-1}) = \frac{C(w_{i-n+1}\cdots w_{i-1}w_i)}{C(w_{i-n+1}\cdots w_{i-1})} \tag{2.24}$$

La formula 2.24 corresponde a la estimación por máxima verosimilitud (LM) donde se maximiza la probabilidad de la muestra [VTdlH$^+$05]. De cualquier manera la formula 2.24 no es adecuada para la estimación del modelo del lenguaje debido a que esta se realiza a partir de una muestra del espacio de eventos, usualmente insuficientemente representativa, y no del espacio de eventos. En muchos casos ocurre que hay n-gramas válidos en un lenguaje que no se han visto nunca en la muestra de entrenamiento, con lo cual cualquier secuencia de palabras que incluya un n-grama no visto tendrá una probabilidad 0. También ocurre que los eventos poco frecuentes en el espacio de eventos, pero que aparecen en la muestra se ven

sobrestimados. Se necesita pues aplicar algún tipo de suavizado. Una forma sencilla de suavizar un n-grama es la interpolación (lineal o no) [Jel98] . Este consiste en interpolar las funciones de frecuencias relativas para el n-grama, el (n-1)-grama, el (n-2)-grama hasta el unigrama. En el caso lineal:

$$\begin{aligned} P(w_i|w_{i-n+1}\cdots w_{i-1}) = \lambda_n f(w_i|w_{i-n+1}\cdots w_{i-1}) + \\ \lambda_{n-1} f(w_i|w_{i-n}\cdots w_{i-1}) + \cdots + \lambda_2 f(w_i|w_{i-1}) + \lambda_1 f(w_i) \end{aligned} \tag{2.25}$$

Donde los pesos λ_i han de ser positivos y se ha satisfacer la restricción de que $\sum_{i=1}^{n} \lambda_i = 1$.

Si la cuenta de un evento es suficientemente grande, la función de frecuencia relativa, $f(|)$ puede ser mejor estimador de la probabilidad que el obtenido mediante interpolación. De tal manera, Katz [Kat87] sugirió que si la frecuencia de un evento era suficiente, se utilizase esta como estimador de la probabilidad y en el caso contrario, es cuando habría que aplicar algún tipo de suavizado.

El método de suavizado propuesto por Katz se llama *Back-off* [Kat87] y como se ha apuntado previamente consiste en utilizar la función de frecuencia relativa si la aparición de un evento supera un cierto umbral K y en caso contrario repartir una pequeña masa de probabilidad entre los eventos poco vistos, o no vistos, pero teniendo en cuenta la proporción de estos en la muestra.

$$P(w_i|w_{i-n}\cdots w_{i-1}) = \begin{cases} f(w_i|w_{i-n+1}\cdots w_{i-1}) & si\ C(w_{i-n+1}\cdots w_i) \geq K \\ \alpha Q_T(w_i|w_{i-n+1}\cdots w_{i-1}) & si\ 1 \leq C(w_{i-n+1}\cdots w_i) < K \\ \beta P(w_i|w_{i-n+1}\cdots w_{i-1}) & en\ otros\ casos \end{cases} \tag{2.26}$$

Donde $Q_T()$ es una función de descuento que reserva una pequeña masa de probabilidad para ser repartida entre los eventos no vistos. Los parámetros α y β han de ser elegidos de manera que aseguren una normalización de la probabilidad del modelo. Para la elección del umbral K no hay definido ningún criterio y ha de ser elegido empíricamente. De la formula 2.26, si la frecuencia de un n-grama en la muestra *es suficientemente alta* se toma la función de frecuencia como estimador de la probabilidad. Si la muestra ha aparecido (*pero no suficientemente*) se utiliza también la función de frecuencia como estimador de la probabilidad, pero rectificada con algún método de descuento. Para los eventos no vistos se le reduce la historia y se invoca a una función equivalente pero para (n-1)-grama. Como se ve esta es una función recursiva cuyo caso base es el unigrama.

En cualquier caso, para mantener la consistencia del modelo de lenguaje se hace necesario hacer una redistribución de la masa de probabilidad de tal manera que se asigne parte de esta a los eventos no vistos. Esta redistribución se hace siguiendo alguna estrategia de descuento de probabilidad de los eventos vistos. Esto es, en vez de utilizar las frecuencias de la muestra r, las sustituye por $rd(r)$ donde $d(r)$ es conocida como la tasa de descuento $0 \leq d(r) \leq 1$. A continuación se describen las estrategias de descuento clásicas.

- **Descuento Good Turing** [Kat87] se define como

$$d(r) = \frac{n_{r+1}}{r n_r}(r+1)$$

donde n_r es el número de eventos que tienen una frecuencia r

- **Descuento Witten-Bell** [WB91] Cuando se habla de este tipo de descuento se suele hacer referencia al definido por Witten y Bell como de tipo C. En este caso, la tasa de descuento depende no sólo de la cuenta de un evento r, sino también de t, el número de tipos de eventos diferentes que tienen un contexto determinado. Este descuento se define como:

$$d(r,t) = \frac{S}{S+t}$$

donde S es el número total de eventos vistos en la muestra con un contexto particular.

- **Descuento Lineal** [NEK94] definido como:

$$d(r) = 1 - \frac{1}{S}$$

donde S es el número total de eventos vistos en la muestra. Este descuento no se aplica solamente a aquellos eventos con una frecuencia mayor que K, sino que por el contrario se aplica a todos aquellos eventos que han sido vistos.

- **Descuento Absoluto** [NEK94] definido como:

$$d(r) = \frac{r-b}{r}$$

donde típicamente $b = \frac{n_1}{n_1+2n_2}$, n_1 y n_2 son el número de eventos que tienen una frecuencia 1 y 2 respectivamente. El Descuento Absoluto se aplica también a todos los eventos que han sido vistos.

2.2.2. Modelos de estados finitos

En este trabajo se han utilizado sistemas basados en una aproximación sin segmentación previa en caracteres ni palabras (”*segmentation free*”). Se han utilizado HMMs para modelar los caracteres. Como el espacio de búsqueda puede resultar intratable, se hace necesario aplicar restricciones para la búsqueda. De ese modo, se definen las secuencias de caracteres que conforman cada palabra y se construye un único modelo HMM para cada palabra a partir de la concatenación de los modelos de caracter que componen la misma. De esa manera, para construir un modelo del lenguaje se han de concatenar las palabras que conforman las frases aceptadas por el lenguaje. De esta manera se construye un gran HMM denominado modelo integrado (ver figura 2.3). Como los HMMs son autómatas de estados finitos,

resulta sencillo modelar cada uno de los otros niveles de conocimiento mediante autómatas de estados finitos. Hay que remarcar que el sistema utilizado en esta tesis no expande todo el modelo a priori, por motivos de coste espacial, sino que lo va expandiendo al *vuelo* según la necesidad.

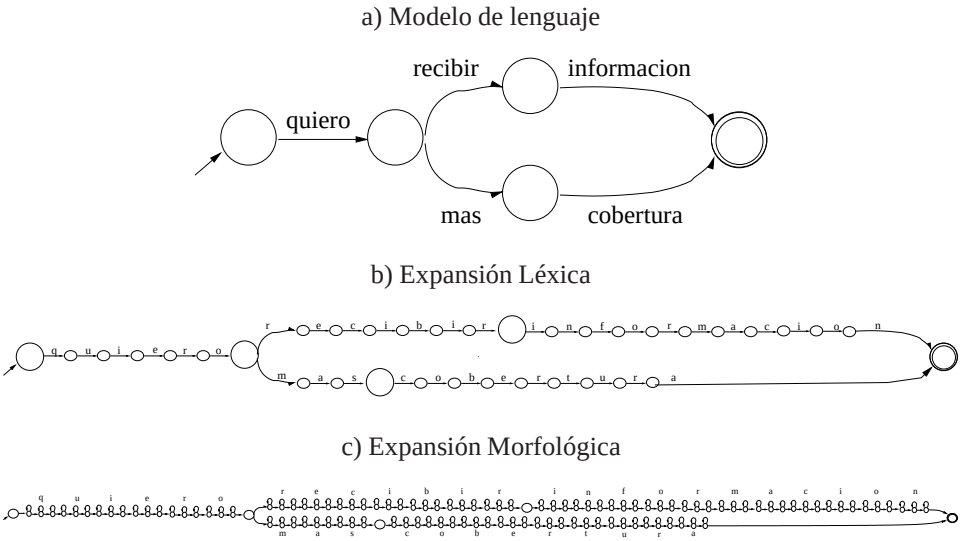


Figura 2.3: Ejemplo de construcción de un modelo integrado.

Una gramática de estados finitos es una tupla $A = (Q, \Sigma, \delta, q_0, F)$ donde:

- Q es un conjunto finito de estados.
- Σ es un conjunto de símbolos, o alfabeto de entrada.
- δ es la función de transición. Dependiendo de como es esta función los autómatas se dividen en deterministas e indeterministas. En los deterministas, a partir de un estado y un símbolo sólo puede ”transitar” a otro estado, es decir $\delta : Q \times \Sigma \longrightarrow Q$. Los indeterministas, por otro lado, a partir de un estado y un símbolo se puede transitar a un conjunto de estados; $\delta : Q \times \Sigma \longrightarrow 2^Q$
- $q_o \in Q$ es el estado inicial.
- $F \subseteq Q$ es un conjunto de estados finales o aceptores.

Los n-gramas pueden ser representados mediante gramáticas deterministas estocásticas. Este tipo de gramáticas se pueden representan mediante una tupla $A = (Q, \Sigma, \delta, q_0, \gamma, \psi)$ donde Q, Σ, δ, q_0 tienen el mismo significado que en el caso anterior, y donde γ es una función que asigna una probabilidad a cada transición de δ, definida como $\gamma : Q \times \Sigma \times Q \rightarrow \Re^+$, y ψ es una función que asigna a cada estado una probabilidad de ser final, definida como $\psi : Q \rightarrow \Re^+$. Estas funciones han de satisfacer las siguientes restricciones:

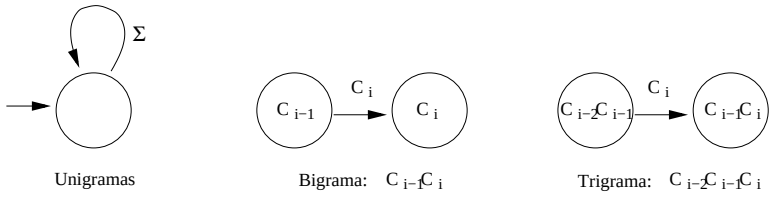


Figura 2.4: Ejemplos de representación de n-gramas mediante gramáticas de estados finitos. Las probabilidades de transición se han obviado por motivos de legibilidad.

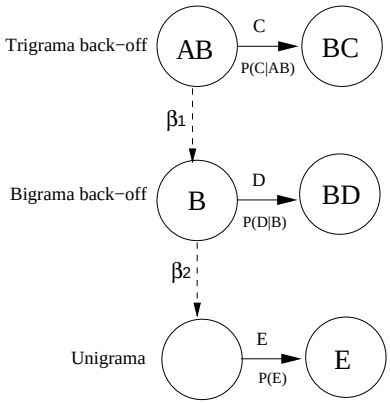


Figura 2.5: Ejemplos de representación de n-gramas suavizados con *back-off* mediante gramáticas de estados finitos.

- $\gamma(q, a, q') = 0; \quad si \quad q' \notin \delta(q, a)$
- $$\sum_{\substack{\forall a \in \Sigma \\ \forall q' \in \delta(q,a)}} \gamma(q, a, q') + \psi(q) = 1, \quad \forall q \in Q$$

En el caso de representar un n-grama mediante una gramática estocástica determinista, es usual etiquetar los estados con la historia con la que se llega a ellos, esto es: $q \subseteq \Sigma^{n-1}$ (ver ejemplos de la figura 2.4). Si el n-grama está suavizado con *back-off*, de cada estado con una historia de longitud $i \geq 1$ se crea una nueva transición vacía que va a un estado donde la historia se ha reducido a longitud $i - 1$. La probabilidad de ese arco es la probabilidad de *back-off* (β) para ese n-grama (ver el ejemplo de la figura 2.5).

2.3. Métrica para la evaluación del sistema de RATM

En esta sección se expone la métricas utilizadas para evaluar las prestaciones de los sistemas de reconocimiento de formas.

2.3.1. Tipos de experimentos

Generalmente para la evaluación de un sistema de RATM se utiliza una porción de las muestras que se tienen de la tarea. Dependiendo de si los escritores que han producido esta porción se han utilizado o no, previamente en el proceso de entrenamiento de los modelos (HMMs, del léxico y del lenguaje) podemos distinguir dos tipos de experimentos.

- Dependiente del escritor. En este caso las muestras para la evaluación del sistema pertenecen al mismo escritor que las utilizadas en el entrenamiento. Este tipo de sistemas producen las mejores tasas de acierto para ese mismo escritor. El problema que surge es que para poder entrenar el sistema se necesitan una gran cantidad de muestras manuscritas de dicho escritor, lo cual es una tarea tediosa.

- Independiente del escritor. Las muestras utilizadas para la evaluación no pertenecen a ningún escritor cuyas muestras hayan sido utilizadas previamente para la estimación de los parámetros de los modelos.

Aunque los sistemas dependientes del escritor producen mejores tasas de acierto, estos son menos interesantes debido a la necesidad de obtener suficientes muestras de entrenamiento del usuario final. Por contra los sistemas independientes del escritor son más genéricos y no requieren ningún esfuerzo por parte del usuario para utilizarlos. La tendencia actual es utilizar una técnica mixta, esto es a partir de un sistema independiente del escritor realizar una rápida adaptación al usuario final. De esta manera se consigue mejorar las prestaciones a cambio de un pequeño esfuerzo por parte del usuario.

2.3.2. SER

Una métrica de evaluación muy usada es la *tasa de error de frase*, SER (del inglés *"Sentence Error Rate"*). El SER se calcula comparando directamente la hipótesis ofrecida por el sistema con una transcripción correcta o "de referencia" de la frase, contabilizando el porcentaje de frases erróneamente clasificadas. Esta es una métrica un poco tosca (dependiendo de la aplicación) ya que cualquier variación entre la hipótesis y la referencia es contada como un error, sin tener en cuenta como de parecidas fuesen las dos frases o el número de palabras de las frases.

2.3.3. WER

Otra métrica muy utilizada es la llamada *tasa de error de palabras*, WER (del inglés *"Word Error Rate"*). El WER ha sido muy utilizado en las últimas décadas para medir las prestaciones de los sistemas automáticos de reconocimiento del habla. Esta métrica contabiliza el número de palabras que difieren entre la hipótesis y la referencia. El WER se calcula mediante un alineamiento entre la hipótesis y la referencia. Nótese que el número de palabras de la hipótesis y la referencia pueden

diferir. En el alineamiento de las dos frases se pueden encontrar cuatro situaciones posibles con respecto a las palabras a alinear.

a) **acierto:** la palabra de referencia y la de la hipótesis alineada coinciden.

b) **sustitución:** La palabra de referencia es alineada a una palabra diferente de la hipótesis.

c) **inserción:** Una palabra de la hipótesis que no ha podido ser alineada con ninguna de la referencia.

d) **borrado:** Una palabra de la referencia que no aparece en la hipótesis.

Las dos frases pueden ser alineadas de muchas maneras, así que habrá que definir alguna manera de alinear que sea "óptima". Se define como alineamiento óptimo a aquel producido por la distancia de edición o de *Levenshtein* [SK83]. La distancia de Levenshtein [Lev66] está definida como el valor mínimo de la suma ponderada de inserciones, borrados, y sustituciones entre las dos secuencias de palabras. Este valor puede obtenerse por programación dinámica [BS89, MG96]. La ponderación más usual es utilizar 1 como peso para los eventos de inserción, borrado y sustitución, mientras que el peso para los aciertos es 0. El WER se calcula pues como el número total de inserciones, borrados y sustituciones (los errores que ha cometido el sistema) obtenidos del alineamiento óptimo de cada frase, dividido por el número total de palabras de la referencia (número de palabras que debería tener la hipótesis si hubiese estado bien reconocida):

$$\text{WER} = \frac{n_b + n_s + n_i}{n_b + n_s + n_a} \tag{2.27}$$

donde:

n_i : es el número de inserciones.
n_b : es el número de borrados.
n_s : es el número de sustituciones.
n_a : es el número de aciertos.

Nótese que el WER puede sobrepasar el $100\,\%$ si el número de inserciones es grande.

CAPÍTULO 3

CORPUS

La utilización de corpus es muy importante para la estimación de los parámetros de los modelos, así como, para la evaluación de los modelos. Los corpus son un requisito indispensable para el desarrollo, la evaluación y la comparación de las diferentes técnicas que se exploren. Como los sistemas explotados en este trabajo están basados en entrenamiento supervisado, se hace necesario no solamente de disponer de las muestras, sino que éstas han de estar etiquetadas.

También es importante disponer de corpus estándares para que los investigadores puedan comparar sus aportaciones con el resto de investigadores. El proceso de recolección de muestras y etiquetado de las mismas es un proceso caro, por lo que no resulta tan sencillo encontrar corpus estándares disponibles.

En este capítulo se exponen las características más importantes de los corpus de texto manuscrito usados en este trabajo.

Consiste en una secuencia de imágenes (matrices bidimensionales) que representan imágenes conteniendo texto. Cada valor de la matriz contiene un valor entero (típicamente es un valor en el rango [0,255]) que representa la cantidad de luz que contendrá ese punto en la imagen. A la densidad de puntos por superficie se la conoce como resolución y hay que hacer notar que a mayor resolución mejor representación del texto se tendrá.

3.1. ODEC[1]

Este es un corpus de escritura manuscrita continua espontánea y con un vocabulario relativamente grande [Tos04, PTV04]. El corpus consiste en una serie de frases espontáneas (ver ejemplos en la figura 3.2) extraídas de formularios de encuestas (ver figura 3.1) para una prestigiosa compañía de telecomunicaciones. Las frases manuscritas estaban contenidas en un apartado de sugerencias. Las frases fueron escritas por un grupo heterogéneo de personas a las que no se impuso ningún tipo de restricción respecto al vocabulario, estilo, estilográfica, etc. Este es un corpus verdaderamente espontáneo donde podemos encontrar un gran número

[1] Datos cedidos por ODEC,S.A. http://www.odec.es

de incorrecciones ortográficas, lógicas y sintácticas, además de un gran número de abreviaciones típicas (o no tanto) de los usuarios de telefonía móvil, tachones y un sinfín de signos no ortográficos. También hay palabras escritas con mayúsculas y minúsculas mezcladas, diferentes tipografías, tamaños, y muestras que incluyen palabras desconocidas o de otros idiomas. La combinación de tan diversos estilos de escritura hace que el preproceso sea un reto importarte para esta tarea.

Este corpus se obtuvo a partir de 2500 formularios de los cuales 950 tenían cumplimentado el apartado de sugerencias. Finalmente, tras descartar algunas de las respuestas debido a que solamente contenían ruido, el corpus quedo conformado con 913 imágenes con un total de $16,325$ palabras y con un vocabulario de 3308 palabras. Los formularios fueron digitalizados a una resolución de 2475×2362 píxeles. Las imágenes se obtuvieron en binario en vez de nivel de gris como es habitual. Las áreas de texto manuscrito fueron extraídas automáticamente el eje de coordenadas con respecto a las cuatro marcas (en forma de cuadrados negros) situadas en los margenes derecho e izquierdo, sobre la mitad inferior de cada formulario. El tamaño del área correspondiente al apartado ocho (el que contiene el texto manuscrito) fue de 1350×560. A esta área quedó reducida mediante la aplicación de una caja de inclusión mínima sobre el texto manuscrito contenido.

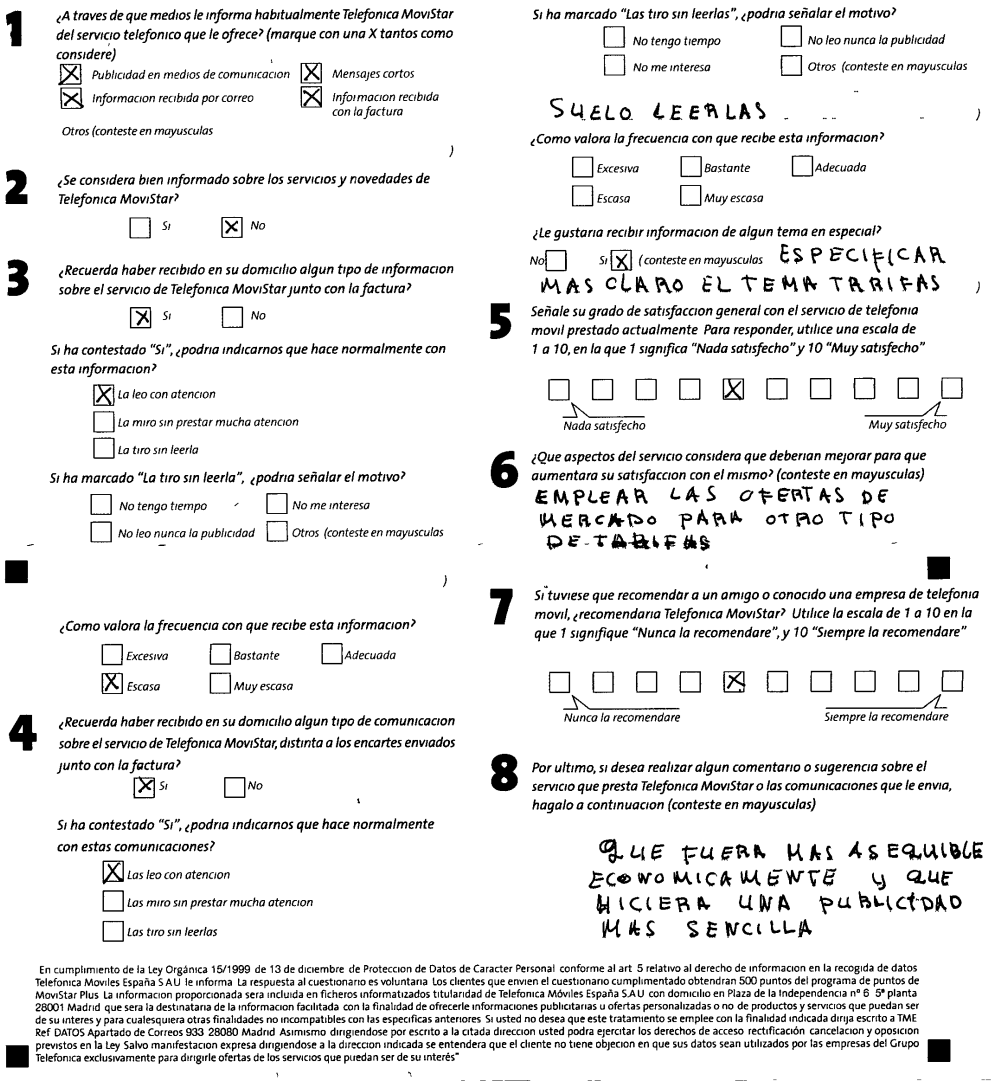
Las lineas se segmentaron manualmente y se dispusieron en un renglón, de tal manera que el párrafo quedaba convertido en una sola línea de texto. El corpus fue particionado en un subconjunto de entrenamiento con un total de 676 imágenes y en un subconjunto de test con 237 imágenes.

El etiquetado o transcripciones de las frases se realizaron manualmente siguiendo la siguiente directiva: describir con el mayor detalle y precisión posible el texto manuscrito. Para lo cual se intentan transcribir las palabras tal cual aparecen en la imagen, con faltas de ortografía, alternancia de mayúsculas y minúsculas, en su lengua original, etc. Se definen códigos para etiquetar los artefactos que aparecen en el texto como por ej. tachones, firmas, subrayados, flechas, etc.

3.2. IAMDB

Este corpus fue compilado por el grupo de investigación *Computer Vision and Artificial Intelligence (FKI)* del instituto *Computer Science and Applied Mathematics (IAM)* de Berna. El corpus de texto manuscrito IAMDB [MB99, MB00, MB01, MB02, ZB00] es de acceso público y gratuito bajo demanda, para propósitos de investigación. El corpus se dio a conocer por primera vez en el ICDAR (International Conference of Document Analysis and Recognition) de 1999 [MB99]. Este corpus es una transcripción manual de parte del corpus *Lancaster Oslo/Bergen Corpus (LOB)* [JLG78] que es una colección de 500 textos en inglés, con aproximadamente unas 2000 palabras cada uno. Los textos del LOB se particionaron en párrafos, conteniendo entres 3 y 6 frases cada uno, con un mínimo de 50 palabras, y se pidió a diferentes personas que lo reescribieran manualmente. No se impuso ninguna

[2] http://www.iam.unibe.ch/∽fki/iamDB



1 ¿A traves de que medios le informa habitualmente Telefonica MoviStar del servicio telefonico que le ofrece? (marque con una X tantos como considere)

- [X] Publicidad en medios de comunicacion
- [X] Mensajes cortos
- [X] Informacion recibida por correo
- [X] Informacion recibida con la factura

Otros (conteste en mayusculas)

2 ¿Se considera bien informado sobre los servicios y novedades de Telefonica MoviStar?

- [ ] Si
- [X] No

3 ¿Recuerda haber recibido en su domicilio algun tipo de informacion sobre el servicio de Telefonica MoviStar junto con la factura?

- [X] Si
- [ ] No

Si ha contestado "Si", ¿podria indicarnos que hace normalmente con esta informacion?

- [X] La leo con atencion
- [ ] La miro sin prestar mucha atencion
- [ ] La tiro sin leerla

Si ha marcado "La tiro sin leerla", ¿podria señalar el motivo?

- [ ] No tengo tiempo
- [ ] No me interesa
- [ ] No leo nunca la publicidad
- [ ] Otros (conteste en mayusculas)

¿Como valora la frecuencia con que recibe esta informacion?

- [ ] Excesiva
- [ ] Bastante
- [ ] Adecuada
- [X] Escasa
- [ ] Muy escasa

4 ¿Recuerda haber recibido en su domicilio algun tipo de comunicacion sobre el servicio de Telefonica MoviStar, distinta a los encartes enviados junto con la factura?

- [X] Si
- [ ] No

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

- [X] Las leo con atencion
- [ ] Las miro sin prestar mucha atencion
- [ ] Las tiro sin leerlas

Si ha marcado "Las tiro sin leerlas", ¿podria señalar el motivo?

- [ ] No tengo tiempo
- [ ] No leo nunca la publicidad
- [ ] No me interesa
- [ ] Otros (conteste en mayusculas

SUELO LEERLAS)

¿Como valora la frecuencia con que recibe esta informacion?

- [ ] Excesiva
- [ ] Bastante
- [ ] Adecuada
- [ ] Escasa
- [ ] Muy escasa

¿Le gustaria recibir informacion de algun tema en especial?

No [ ] Si [X] (conteste en mayusculas ESPECIFICAR MAS CLARO EL TEMA TARIFAS)

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

[ ] [ ] [ ] [ ] [X] [ ] [ ] [ ] [ ] [ ]

Nada satisfecho — Muy satisfecho

6 ¿Que aspectos del servicio considera que deberian mejorar para que aumentara su satisfaccion con el mismo? (conteste en mayusculas)

EMPLEAR LAS OFERTAS DE MERCADO PARA OTRO TIPO DE TARIFAS

7 Si tuviese que recomendar a un amigo o conocido una empresa de telefonia movil, ¿recomendaria Telefonica MoviStar? Utilice la escala de 1 a 10 en la que 1 signifique "Nunca la recomendare", y 10 "Siempre la recomendare"

[ ] [ ] [ ] [ ] [X] [ ] [ ] [ ] [ ] [ ]

Nunca la recomendare — Siempre la recomendare

8 Por ultimo, si desea realizar algun comentario o sugerencia sobre el servicio que presta Telefonica MoviStar o las comunicaciones que le envia, hagalo a continuacion (conteste en mayusculas)

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

En cumplimiento de la Ley Orgánica 15/1999 de 13 de diciembre de Proteccion de Datos de Caracter Personal conforme al art 5 relativo al derecho de informacion en la recogida de datos Telefonica Moviles España S A U le informa La respuesta al cuestionario es voluntaria Los clientes que envien el cuestionario cumplimentado obtendran 500 puntos del programa de puntos de MoviStar Plus La informacion proporcionada sera incluida en ficheros informatizados titularidad de Telefonica Móviles España S A U con domicilio en Plaza de la Independencia nº 6 5ª planta 28001 Madrid que sera la destinataria de la informacion facilitada con la finalidad de ofrecerle informaciones publicitarias u ofertas personalizadas o no de productos y servicios que puedan ser de su interes y para cualesquiera otras finalidades no incompatibles con las especificas anteriores Si usted no desea que este tratamiento se emplee con la finalidad indicada dirija escrito a TME Ref DATOS Apartado de Correos 933 28080 Madrid Asimismo dirigiendose por escrito a la citada direccion usted podra ejercitar los derechos de acceso rectificación cancelacion y oposicion previstos en la Ley Salvo manifestacion expresa dirigiendose a la direccion indicada se entendera que el cliente no tiene objecion en que sus datos sean utilizados por las empresas del Grupo Telefonica exclusivamente para dirigirle ofertas de los servicios que puedan ser de su interés"

Figura 3.1: Ejemplo de formulario de encuesta ODEC

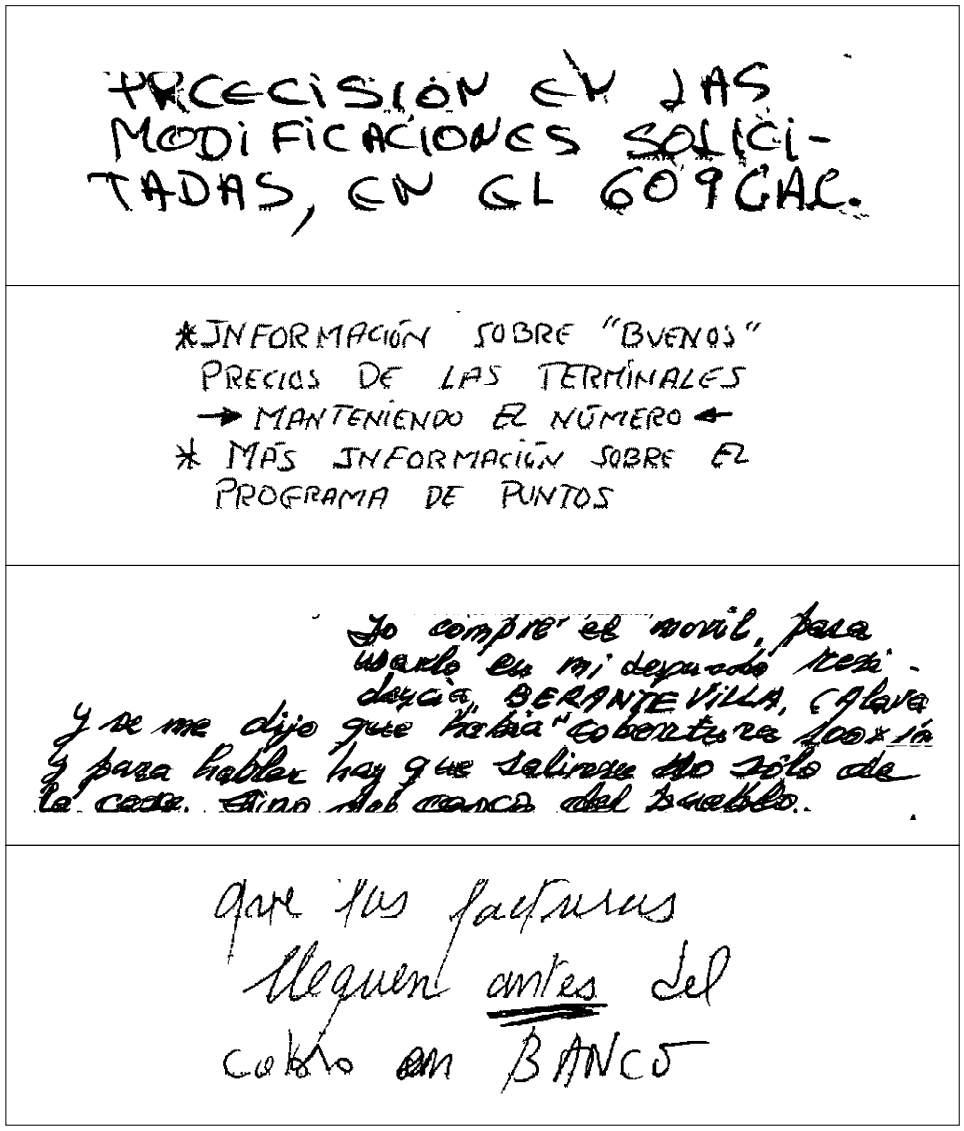


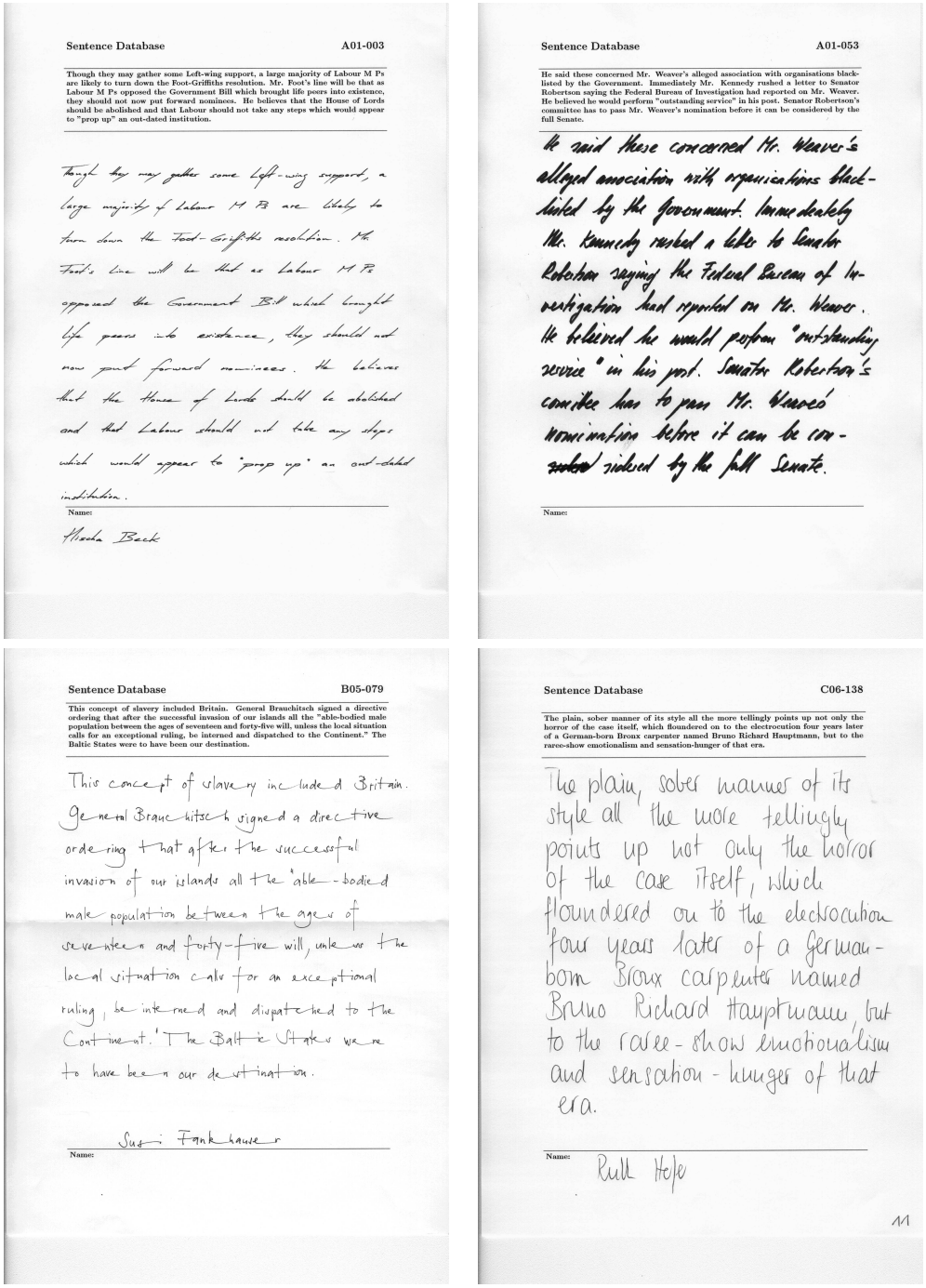
Figura 3.2: Diversos ejemplos de texto extraídas de las casillas de sugerencias de los formularios de las encuestas de ODEC.

restricción en cuanto al tipo de estilográfica a utilizar o al estilo de escritura. Este es por lo tanto, aunque el texto sea predeterminado, un texto espontáneo en cuanto al estilo de escritura. El corpus Iamdb está segmentado tanto a nivel de palabra como de frase. En este trabajo se hará uso del corpus segmentado a nivel de frase. En la figura 3.3 se pueden ver algunos ejemplos de páginas manuscritas de este corpus.

Los recopiladores de este corpus definieron una tarea basada en léxico cerrado e independiente del escritor. Esta tarea consta de una partición para entrenamiento de 2124 frases con un número de palabras alrededor de las 43.000, y una partición para test compuesta por 200 frases con un total de aproximadamente 4.000 palabras. El vocabulario está formado por unas 8500 palabras. Hay que remarcar que para cada palabra existen diversas formas de escribirla dependiendo sobretodo de la utilización de las máyusculas y mínusculas. Si se toma las posibles formas de escribir cada palabra, el vocabulario asciende a aproximadamente 11.000 palabras.

Características	Odec	Iamdb
escritores	913	657
palabras	16.325	46.789
vocabulario	3.308	8.497
lineas de texto	913	2.324
palabras entrenamiento	12.137	42.832
palabras test	4.188	3.957

Tabla 3.1: Resumen de los corpus de texto manuscrito off-line



Sentence Database A01-003

Though they may gather some Left-wing support, a large majority of Labour M Ps are likely to turn down the Foot-Griffiths resolution. Mr. Foot's line will be that as Labour M Ps opposed the Government Bill which brought life peers into existence, they should not now put forward nominees. He believes that the House of Lords should be abolished and that Labour should not take any steps which would appear to "prop up" an out-dated institution.

Though they may gather some Left-wing support, a large majority of Labour M Ps are likely to turn down the Foot-Griffiths resolution. Mr. Foot's line will be that as Labour M Ps opposed the Government Bill which brought life peers into existence, they should not now put forward nominees. He believes that the House of Lords should be abolished and that Labour should not take any steps which would appear to "prop up" an out-dated institution.

Name:

Hasha Beck

Sentence Database A01-053

He said these concerned Mr. Weaver's alleged association with organisations blacklisted by the Government. Immediately Mr. Kennedy rushed a letter to Senator Robertson saying the Federal Bureau of Investigation had reported on Mr. Weaver. He believed he would perform "outstanding service" in his post. Senator Robertson's committee has to pass Mr. Weaver's nomination before it can be considered by the full Senate.

He said these concerned Mr. Weaver's alleged association with organisations blacklisted by the government. Immediately Mr. Kennedy rushed a letter to Senator Robertson saying the Federal Bureau of Investigation had reported on Mr. Weaver. He believed he would perform "outstanding service" in his post. Senator Robertson's committee has to pass Mr. Weaver's nomination before it can be considered by the ~~full~~ blacklisted by the full Senate.

Name:

Sentence Database B05-079

This concept of slavery included Britain. General Brauchitsch signed a directive ordering that after the successful invasion of our islands all the "able-bodied male population between the ages of seventeen and forty-five will, unless the local situation calls for an exceptional ruling, be interned and dispatched to the Continent." The Baltic States were to have been our destination.

This concept of slavery included Britain. General Brauchitsch signed a directive ordering that after the successful invasion of our islands all the "able-bodied male population between the ages of seventeen and forty-five will, unless the local situation calls for an exceptional ruling, be interned and dispatched to the Continent.' The Baltic States were to have been our destination.

Name:

Susi Frankhauser

Sentence Database C06-138

The plain, sober manner of its style all the more tellingly points up not only the horror of the case itself, which floundered on to the electrocution four years later of a German-born Bronx carpenter named Bruno Richard Hauptmann, but to the raree-show emotionalism and sensation-hunger of that era.

The plain, sober manner of its style all the more tellingly points up not only the horror of the case itself, which floundered on to the electrocution four years later of a German-born Broux carpenter named Bruno Richard Hauptmann, but to the raree-show emotionalism and sensation-hunger of that era.

Name:

Ruth Hofe

11

Figura 3.3: Ejemplo de páginas escaneadas del corpus IAMDB

CAPÍTULO 4

PREPROCESO

Una de las principales dificultades con la que se enfrenta el reconocimiento de texto manuscrito continuo es la gran diversidad de estilos de escritura. Esta diversidad de estilos se presenta no sólo entre diferentes escritores, sino también en cada escritor dependiendo de su estado de ánimo, velocidad a la que escribe y de la atención prestada al escribir. Otros factores como la utilización de diferentes instrumentos de escritura contribuyen a dicha dificultad.

El estilo de escritura no aporta nada a los sistemas de reconocimiento de texto manuscrito (salvo para sistemas biométricos [RK98]), ya que lo escrito no tiene nada que ver con el estilo en que está escrito. Además, los modelos matemáticos utilizados para representar la escritura, y los métodos que los estiman, tienen una expresividad limitada, e interesa no desperdiciarla en aspectos que no ayudan a reconocer lo que hay escrito.

El preproceso consiste en una serie de transformaciones sobre la señal original con la intención de obtener la máxima homogeneidad posible dentro de cada clase. La intención es hacer el sistema invariante a las fuentes de variabilidad que no ayuden a la clasificación. De hecho lo que se pretende es que el módulo de extracción de características produzca vectores de características lo más parecidos entre sí para patrones de una misma clase. O dicho de otra manera hace que el sistema sea robusto frente a la entrada de texto manuscrito.

La necesidad de incluir esta etapa en los sistemas de reconocimiento de patrones, en general, ha sido demostrada empíricamente a lo largo de años de estudio, y hoy en día los esquemas generales de reconocimiento de formas incluyen este módulo.

En la actualidad no hay definida una solución general para conseguir invariabilidad al estilo de escritura, y cada sistema desarrolla la suya *ad-hoc*. Sin embargo hay algunas características bien conocidas que definen, en parte, al estilo de escritura, y cuya normalización es tenida en cuenta por la práctica totalidad de los sistemas actuales: Entre los más conocidos cabe destacar el *slope* que es la desviación de cada palabra con respecto a la horizontal, el *slant*, ángulo del las componentes verticales del texto respecto al eje vertical, y el tamaño de los caracteres.

Los métodos de preproceso son difíciles de evaluar de forma independiente

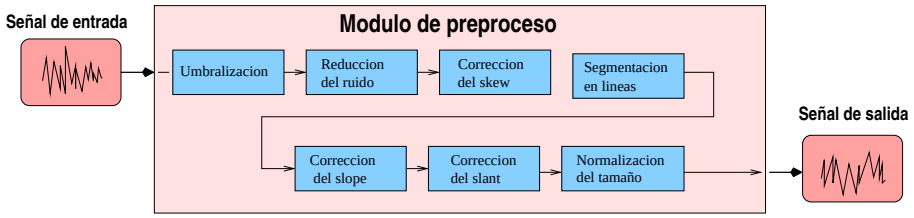


Figura 4.1: Esquema general de preproceso *off-line*.

debido a que existe correlación entre los diferentes módulos del sistema. La bondad de los métodos ha de ser evaluada con respecto al sistema global [TJ95].

Dependiendo de la naturaleza de la entrada del sistema podemos hablar de dos tipos de preproceso: *on-line* u *off-line*. En este trabajo nos vamos a centrar en el preproceso *off-line*. Lo que caracteriza al preproceso *off-line* en contraste con el *on-line* es la manera en que se obtiene el texto. Este se adquiere de manera indirecta, primero se escribe en papel o cualquier soporte físico, para luego ser digitalizado utilizando un escáner o una cámara. El texto *on-line* se obtiene directamente mediante algún dispositivo que muestree la trayectoria de un móvil.

Una imagen bidimensional, $f(x, y)$, es una función, $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, que para cada punto del espacio bidimensional devuelve el nivel de luz de dicho punto. Una imagen *off-line* es una función, $I(x, y)$, donde el espacio bidimensional y los niveles de gris se han discretizado; $I : \{1, \cdots, F\} \times \{1, \cdots, C\} \rightarrow \{1, \cdots, L\}$, donde L suele ser 255, y F,C son las filas y columnas de la matriz. A cada una de las celdas de esta matriz se la denomina píxel (es la forma abreviada de las palabras **pix***ture* **el***emen*). El proceso de adquisición consiste pues en una doble discretización: del espacio y de los niveles de gris, o dicho de otro modo, es el proceso que permite tranformar una imagen bidimensional en una imagen *off-line*

En el presente capítulo se describen los métodos utilizados en este trabajo para el preproceso de las imágenes *off-line*. El capítulo se divide en dos grandes bloques. En el primero se tratarán técnicas de normalización a nivel de página, mientras que en la segunda parte se normalizará el estilo de escritura. En la primera parte, la sección 4.1.1 trata la umbralización de las imágenes codificadas en niveles de gris. En la sección 4.1.2 se exponen las técnicas utilizadas en este trabajo para la minimización del ruido existente en las imágenes. En la sección 4.1.3 se exponen todas las técnicas utilizadas para corregir el desencuadre de página o *skew*. En el bloque de normalización del estilo de escritura, la sección 4.2.1 explica como corregir la inclinación de la línea base de una palabra o un conjunto de ellas. En la sección 4.2.2 se expone como corregir la inclinación de los caracteres. Por último, la sección 4.2.3 trata de la normalización de la altura de los caracteres.

4.1. Normalización a nivel de página

Todos los métodos expuestos en este bloque actuarán sobre la página completa. Se tratan técnicas de umbralización muy usadas para simplificar la imagen. Así como técnicas para intentar reducir el ruido que pueda llevar consigo la imagen. Como ruido se entiende toda aquella señal parasitaria añadida a la señal que codifica la información.

4.1.1. Umbralización

La umbralización, o *thresholding* consiste en transformar una imagen digital I en escala de grises en una imagen digital binaria. Los métodos de umbralización clasifican los puntos de la imagen en dos clases: los que pertenecen al fondo de la imagen o segundo plano, *background pixels* y los que pertenecen al texto o primer plano, *foreground pixels*. Esta clasificación está basada en la elección de un umbral que divide los puntos en dos clases, aquellos cuyo valor está por encima del umbral $T_{i,j}$ que se clasificarán como *background pixels* (píxeles con valores elevados codifican tonos más claros) y los que están por debajo como *foreground pixels* (formula 4.1). Determinar un umbral conveniente no es una tarea trivial. Teniendo en cuenta como se elige el umbral, los métodos de umbralización se clasifican en dos grandes grupos: métodos globales y métodos adaptativos o locales.

$$g(i,j) = \begin{cases} 0, & I(i,j) \geq T_{i,j} \\ 1, & I(i,j) < T_{i,j} \end{cases} \tag{4.1}$$

En binario se suele codificar los píxeles *foreground* (negros) como 1 mientras que los de *background* como 0.

Métodos globales

Los métodos globales calculan un solo umbral para toda la imagen. Usualmente estos métodos son más simples y rápidos, aunque por otra parte no se adaptan bien si el ruido no es regular, o si la iluminación no es uniforme.
En la literatura se pueden encontrar diversos métodos de umbralización globales, aunque el más conocido y usado es el de Otsu [Ots79].

- **Umbralización Otsu**: Este método formula el problema de encontrar un umbral como un problema de optimización. Este método pertenece a la familia de los algoritmos que maximizan la varianza del nivel de gris entre la clase *background* y *foreground*, mientras que minimiza la varianza dentro de cada clase.

  Primero se calcula el histograma normalizado. $p_i = n_i/N$ donde n_i es el número de píxeles con un nivel de gris i, mientras que N representa el número total de píxeles. De tal manera que $p_i \in [0:1]$ y $\sum_{i=0}^{L} p_i = 1$ donde L es el número total de niveles de gris en histograma. Se dividen los píxeles en

dos clases C_0 y C_1 utilizando un umbral k. Esto es, todos los píxeles con un nivel de gris entre 1 y k se pertenecerán a la clase C_0 mientras los que tengan un nivel entre $k+1$ y L pertenecerán a la clase C_1. La probabilidad de que un píxel cualquiera de la imagen pertenezca a una clase viene determinada por las ecuaciones 4.2.

$$P(C_0) = \sum_{i=1}^{k} p_i \qquad P(C_1) = \sum_{i=k+1}^{L} p_i = 1 - P(C_0) \tag{4.2}$$

La probabilidad de que dada una clase, un nivel de gris pertenezca a dicha clase viene definida por:

$$P(i|C_0) = \frac{p_i}{\sum_{j=1}^{k} p_j} = \frac{p_i}{P(C_0)} \qquad P(i|C_1) = \frac{p_i}{\sum_{j=k+1}^{L} p_j} = \frac{p_i}{1-P(C_0)} \tag{4.3}$$

La media de cada clase se calcularía según se define en las formulas 4.4.

$$\begin{aligned} \mu_0 &= \sum_{i=1}^{k} iP(i|C_0) &&= \tfrac{1}{P(C_0)} \sum_{i=1}^{k} ip_i \\ \mu_1 &= \sum_{i=k+1}^{L} iP(i|C_1) &&= \tfrac{1}{1-P(C_0)} \sum_{i=k+1}^{L} ip_i \end{aligned} \tag{4.4}$$

Por lo tanto la varianza de cada clase vendrá definida por las ecuaciones 4.5.

$$\begin{aligned} \sigma_0^2 &= \sum_{i=1}^{k} (i-\mu_0)^2 P(i|C_0) &&= \tfrac{1}{P(C_0)} \sum_{i=1}^{k} p_i \, (i-\mu_0)^2 \\ \sigma_1^2 &= \sum_{i=k+1}^{L} (i-\mu_1)^2 P(i|C_1) &&= \tfrac{1}{1-P(C_0)} \sum_{i=k+1}^{L} p_i \, (i-\mu_1)^2 \end{aligned} \tag{4.5}$$

La varianza global viene definida por la expresión:

$$\sigma_T^2 = \frac{1}{L} \sum_{i=1}^{L} (i-\mu_T)^2 \tag{4.6}$$

donde μ_T es el nivel medio de gris de toda la imagen. La varianza intraclase se define como:

$$\sigma_W^2 = P(C_0)\sigma_0^2 + P(C_1)\sigma_1^2 \tag{4.7}$$

mientras que la varianza interclase se define como:

$$\begin{aligned}\sigma_B^2 = \quad & P(C_0)(\mu_0 - \mu_T)^2 + P(C_1)(\mu_1 - \mu_T)^2 = \\ & P(C_0)P(C_1)(\mu_0 - \mu_1)^2\end{aligned} \tag{4.8}$$

Se deberá encontrar un valor para k de tal manera que maximice alguna de las siguientes funciones objetivo.

$$\eta_0 = \frac{\sigma_B^2}{\sigma_W^2} \qquad \eta_1 = \frac{\sigma_T^2}{\sigma_W^2} \qquad \eta_3 = \frac{\sigma_B^2}{\sigma_T^2} \tag{4.9}$$

Otsu opta por maximizar la función η_3. Como σ_T^2 no depende del parámetro k, maximizar esta función es equivalente a maximizar su numerador. Así pues, el cálculo del máximo para la función objetivo se reduce a calcular el máximo para σ_B^2

$$\hat{k} = \underset{1 \leq k \leq L}{\text{argmax}}\ \sigma_B^2(k) \tag{4.10}$$

- **Umbralización** *Global selection threshold* [KB06]. A partir de un umbral inicial $U_o = \hat{k}$, obtenido a partir del algoritmo de Otsu, se prueba a umbralizar con cada umbral del rango $[U_0 : 255]$. Para cada una de las imágenes obtenidas, se obtiene un grafo, donde cada vértice representa un píxel negro, y los arcos representan la relación de vecindad (usualmente 8-conectados). De cada grafo se obtienen sus componentes conexas. El umbral que produzca la imagen binarizada con menor número de componentes conexas, será tomado como óptimo (figuras 4.2, 4.3 y 4.4).

Métodos adaptativos o locales

En algunas situaciones en las que la iluminación es desigual, con baja resolución, en documentos con poca calidad o con estructuras complejas, un único umbral puede no ser suficiente. Los esquemas locales calculan un umbral para cada píxel, o por cada pequeña región, basándose en la información de los píxel del área local. Aquí se presentan algunos de los métodos de umbralización más populares.

- **Umbralización Bernesen** [Ber86]. Para cada píxel (x, y) este método calcula un umbral $U(x, y) = (Z_{min} + Z_{max})/2$ donde Z_{min} y Z_{max} son el máximo y mínimo valor de los píxeles contenidos en una ventana de tamaño $s \times s$ centrada en el píxel (x, y), respectivamente. Por otra parte, si el contraste en la ventana de análisis ($Z_{max} - Z_{min}$) es menor que un umbral l, el píxel se etiqueta como *background*, ya que la variabilidad dentro de una ventana conteniendo texto ha de ser alta.

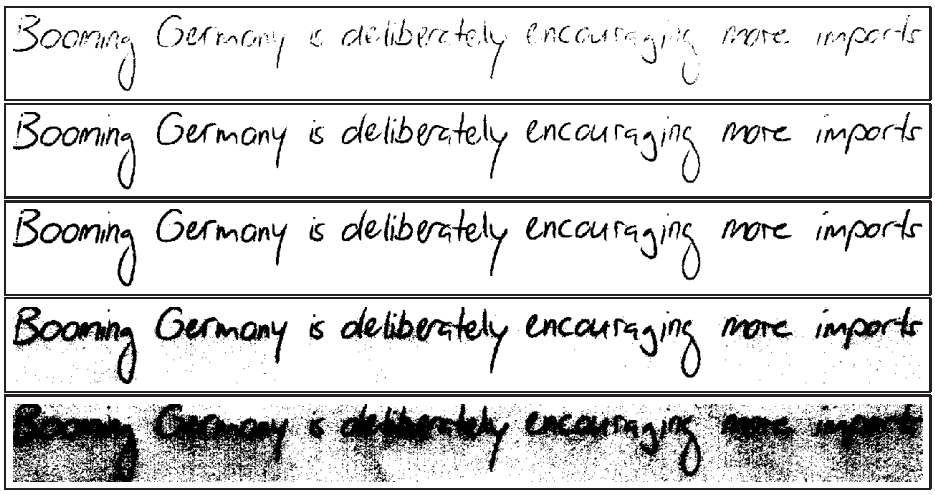


Figura 4.2: Ejemplos de imagenes umbralizadas con diferentes umbrales. De arriba a abajo: 150, 197 (umbral otsu), 220, 240, 250. Se puede apreciar que tal como se incrementa el umbral, el número de componentes conexas crece, y que conforme se va decrementando el umbral, el texto contenido en la imagen se va fragmentando, aumentando también el número de componentes conexas.

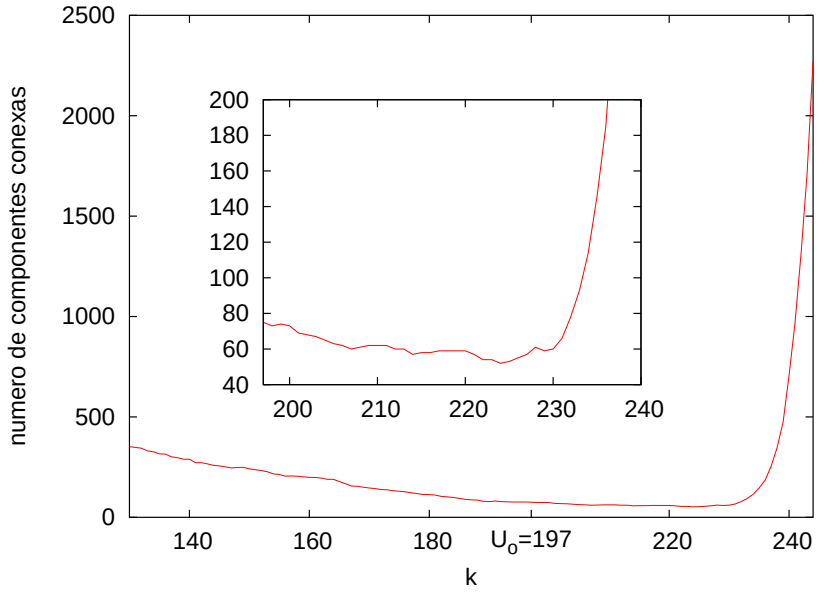


Figura 4.3: Distribución del número de componentes conexas con respecto al umbral elegido. El umbral U_0 es el umbral obtenido con el método de Otsu.

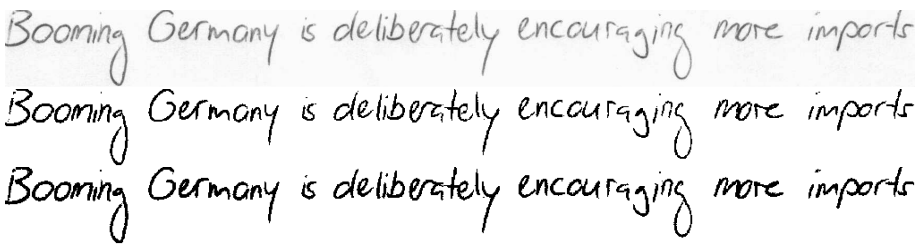


Figura 4.4: Ejemplo de umbralización. Arriba la imagen original, en el centro, la imagen umbralizada con el algoritmo de Otsu, abajo la imagen umbralizada con el método *Global selection threshold*.

- **Umbralización Eikvil** [ETM91]. La imagen se divide en celdas (S) de un tamaño $s \times s$. A cada celda se le superpone una ventana (L) centrada en S, de tamaño $l \times l$ tal que $l > s$. Se utiliza un método global para clasificar los píxeles de la ventana L en la clase *background* o *foreground*. Se obtiene la media de las 2 clases μ_b, μ_f. Si la distancia entre las dos medias es mayor que un umbral dado $|\mu_b - \mu_f| \geq U$, los píxeles de la celda S se binarizan utilizando el método global, en caso contrario, cada píxel se clasifica en la clase a cuya media este mas cercano. Este método no resulta demasiado sensible al valor de la U, aunque es necesario tomar un valor de l suficientemente grande. El método de Eikvil es una mejora del de Taxt [TFJ89]. En el método de Taxt, el histograma de niveles de gris de cada celda es aproximado por una mezcla de dos gausianas. Los parámetros de las gausianas son estimados utilizando el muy conocido algoritmo EM. Los píxeles de cada ventana son clasificados utilizando el clasificador cuadrático de Bayes. Este método, a parte de ser muy costoso computacionalmente, realiza cambios abruptos en la frontera entre celdas. Para evitar cambios abruptos Eikvil utiliza dos ventanas superpuestas, y utiliza el algoritmo de Otsu como una aproximación de EM.

- **Umbralización Niblack** [Nib86]. Este método calcula un umbral basado en medias y desviaciones estándar locales. Cada umbral se calcula de la siguiente manera $U(x,y) = \mu(x,y) + w \cdot \sigma(x,y)$ donde $\mu(x,y)$ y $\sigma(x,y)$ son la media y la desviación típica calculadas sobre una ventana centrada en el píxel (x,y). Este método tiene dos parámetros, el tamaño de la ventana de análisis y el peso w. El método se muestra muy sensible al tamaño de la ventana, el cual es determinante. Debe ser suficientemente pequeño como para preservar los detalles de la imagen y ha de ser suficientemente grande para que se elimine el ruido. El parámetro w se utiliza para determinar el porcentaje de la frontera entre el fondo y el texto que será tomado como parte de la imagen. Si w es positivo la mayor gran parte de la frontera entre el texto y el fondo pasa a ser texto (pasan a ser píxeles negros), mientras que cuando w

es negativo se va a fondo.

- **Umbralización Sauvola** [SP00]. Este método es una variante del de Niblack. Sauvola parte de la hipótesis de que las imágenes de texto manuscrito los píxeles del *background* toman valores cercanos a 255 mientras que los pertenecientes al *foreground* toman valores cercanos a 0. En este caso el umbral se calcula teniendo en cuenta el rango dinámico de la desviación típica R. La media es utilizada para multiplicar los términos R y un valor fijo de $k \in [0:1]$, esto produce una amplificación del efecto de la desviación típica, de manera adaptativa. El valor de cada umbral se calcula de la siguiente manera:

$$U(x,y) = \mu(x,y)\left((1-w) + w\,\frac{\sigma(x,y)}{R}\right)$$

donde $\mu(x,y)$ y $\sigma(x,y)$ se calculan de la misma manera que en el caso de Niblak. Sauvola propone valores de $R = 128$ para niveles de grises de 256 y $w = 0{,}5$. El algoritmo, a diferencia del de Niblak, no resulta tan sensible al parámetro w .

4.1.2. Reducción del ruido

Como ruido se entiende aquella señal añadida a la señal originaria que no aporta ninguna información. Asumiendo que el ruido es puramente aditivo, la imagen original $f(i,j)$ puede ser reescrita de la siguiente manera:

$$f(i,j) = s(i,j) + n(i,j) \tag{4.11}$$

donde $s(i,j)$ sería la imagen ideal y $n(i,j)$ sería una imagen de ruido puro. El problema es como estimar $s(i,j)$ a partir de la imagen compuesta $f(i,j)$ [DH74].

Los métodos más usados en la literatura para la eliminación de estas señales parasitarias están basados en teoría de filtros y en morfología matemática. La mayoría de estos filtros son convoluciones. Convolución es la suma ponderada del contexto para cada píxel de la imagen (ver ecuación 4.12). Para ello, se obtiene una submatriz formada por el contexto alrededor del píxel, el nivel de gris de cada píxel de la submatriz se pondera. Los pesos utilizados para ponderar se representan en una submatriz del mismo tamaño que el contexto. A esta matriz de ponderación se la conoce como matriz de convolución o *kernel*. Una vez ponderado cada píxel del contexto, se suman para obtener el valor de gris del píxel correspondiente en la imagen convolucionada.

$$O(i,j) = I(i,j) \otimes K(i,j) = \sum_{k=\frac{-m}{2}}^{m/2} \sum_{l=\frac{-n}{2}}^{n/2} I(i+k, j+l)K(k,l) \tag{4.12}$$

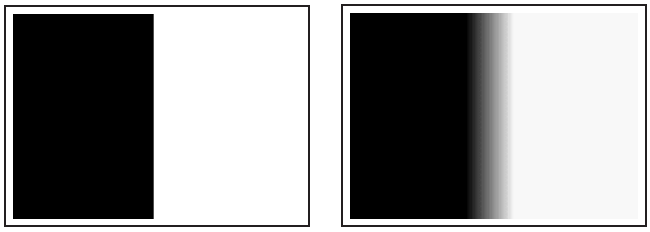


Figura 4.5: Detalle de aplicación de un filtro media. La frontera entre la zona blanco y la negra se suaviza, produciéndose una gradación de tono.

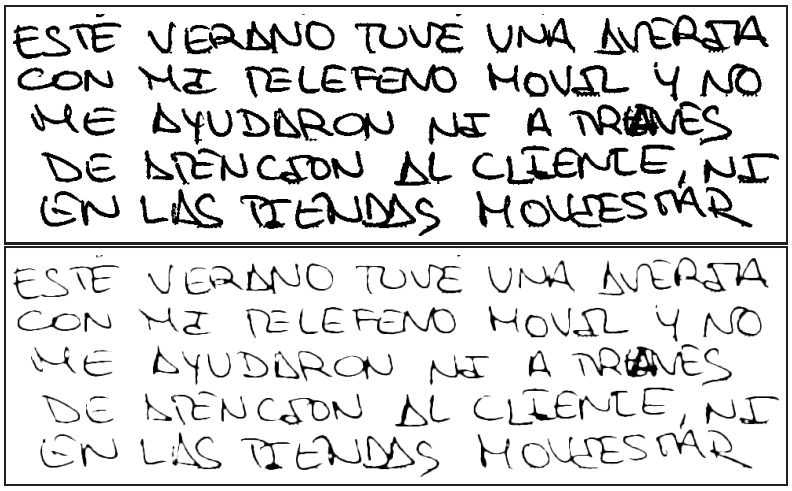


Figura 4.6: Ejemplo de texto preprocesado con filtro media. Arriba la imagen original, abajo la imagen procesada utilizando un kernel de 5×5.

- Filtro mediana. Para cada píxel de la imagen se calcula la mediana de los píxeles contenidos en una ventana centrada sobre él. El valor del píxel se substituye por el valor de la mediana. Este filtro es eficiente en la eliminación de ruido del tipo sal y pimienta. El tamaño de la ventana de contexto es determinante para eliminar el ruido sin degradar demasiado la imagen. Si el tamaño de la ventana, o *kernel*, es suficientemente grande, las altas frecuencias (las que corresponden a cambios abruptos, como por ejemplo el texto) se eliminan. Así pues, los filtros mediana pueden ser vistos como filtros pasobajo. Basándose en esta idea, los filtros mediana se suelen utilizar para obtener una imagen del fondo, la cual puede substraerse en un siguiente paso a la imagen primigenia obteniendo una segunda imagen sin el fondo (ver figura 4.7).

- Filtro media. Este filtro substituye el valor de cada píxel por la media de los píxeles de una ventana centrada sobre él (ver figura 4.6). El efecto de este filtro es el de suavizado de los bordes entre el *background* y el *foreground* (ver figura 4.5).

- Reducción basada en grafos. Estos métodos se suelen utilizar para postprocesar imágenes umbralizadas. Después de una umbralización suele aparecer ruido debido a la clasificación de píxeles del fondo como píxeles de texto. La imagen binaria se representa mediante un grafo donde cada vértice corresponde con un píxel y los arcos explican la relación de vecindad (usualmente 8-conectados). Se obtienen las zonas conexas del grafo y el número de vértices (píxeles) que la conforman [DH74]. De esta manera sólo las zonas conexas con un número menor de vértices que un valor de corte dado se reetiquetarán como píxeles de *background*. Los puntos de corte se eligen de manera que sobreviva un porcentaje del total de la masa de píxeles de *foreground*. Para elegir el punto de corte, se ordenan las zonas conexas por cantidad de vértices, se van acumulando de mayor a menor hasta que se sobrepase el porcentaje deseado, el número de vértices de la última zona conexa se tomará como valor de corte.

  Este método tiene el problema de que los caracteres aislados o las discontinuidades en el trazo pueden provocar la perdida de parte del texto. Con el fin de evitar este efecto la imagen binarizada se procesa con el algoritmo de suavizado por longitud de píxeles horizontales consecutivos *Run-Lengh Smoothing Algorithm* (**RLSA**) [WCW82]. Este algoritmo cambia los píxeles de fondo por píxeles de imagen para aquellos tramos horizontales que estén comprendidos entre dos píxeles de imagen y el número de píxeles de fondo seleccionados sea menor que un umbral establecido previamente (ver algoritmo 1). El umbral suele elegirse empíricamente. Tras este proceso se aplica el algoritmo de reducción basada en grafos.

  Tras este proceso se obtiene una máscara que se superpone a la imagen original para obtener un texto limpio. La utilización previa del algoritmo RLSA proporciona una gran robustez a este método.

Algoritmo 1 RLSA: Run-Lengh Smoothing Algorithm

```
1: Input: imagen, umbral.
2: Output: imagen.
3: for all línea de la imagen do
4:   for all tramos de píxeles de fondo consecutivos entre píxeles de imagen do
5:     if número de píxeles del tramo < umbral then
6:       cambiar los píxeles del tramo a píxeles de imagen
7:     end if
8:   end for
9: end for
```

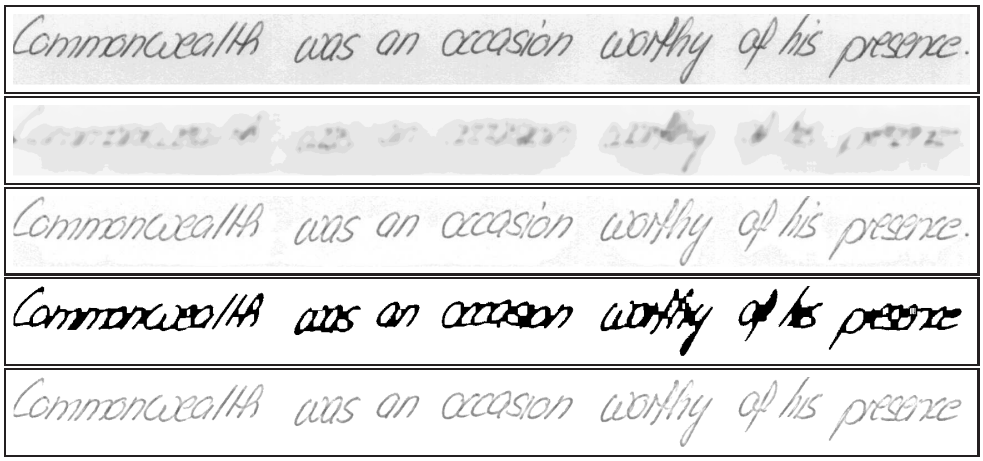


Figura 4.7: Ejemplo de reducción de ruido. De arriba a abajo: imagen original; fondo extraído de la imagen original con un filtro mediana; resta de las dos imágenes anteriores; máscara obtenida de la aplicación del algoritmo RLSA sobre la imagen anterior, más un postproceso de limpieza mediante componentes conexas, situando el punto de corte de manera que se conserve el 98 % de los píxeles negros; resultado de aplicar la mascara.

4.1.3. Corrección del *skew* o desencuadre

El desencuadre es una forma de ruido introducido al escanear el documento, y consiste en la falta de alineamiento del documento de papel con respecto a las coordenadas del escáner utilizado para su digitalización. La corrección del desencuadre facilita la extracción de párrafos y lineas de texto de los documentos. La extracción de frases a partir del texto es muy difícil, sino imposible, si no se reconoce previamente el texto, por lo que el módulo segmentador suele devolver el texto segmentado por lineas, sin tener en cuenta si corresponden a una frase, a parte de una frase, o a varias frases.

Algunos autores suelen referirse al desencuadre como *skew* [AF00, Bun03, CWL03, Hul98, HB04, JBWK99, SS97], otros definen *skew* como una falta de alineamiento de una o varias palabras sobre el eje de abscisas, lo que en este trabajo definiremos como *slope* [GB04, MB01, MFB$^+$99, SRI99, Vin02] y algunos autores lo utilizan indistintamente para referirse al desencuadre de toda la página, como para la mal alineación de palabras con el eje de abscisas.

Una vez se ha estimado el ángulo de desencuadre α, su corrección consiste en aplicar una operación de rotación (ecuación 4.13) con el mismo ángulo en sentido contrario.

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} = \begin{pmatrix} cos(\alpha) & sin(\alpha) \\ -sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} x_{central} \\ y_{central} \end{pmatrix} \qquad (4.13)$$

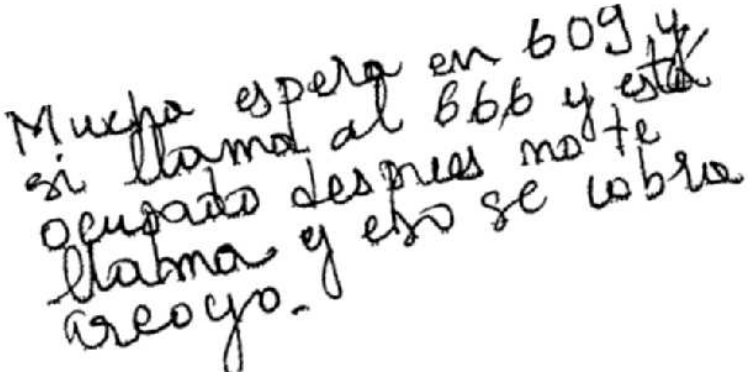


Figura 4.8: Ejemplo de texto con mucho desencuadre

Visto de otra manera, se considera que la imagen con *skew* es el resultado de aplicar a la imagen ideal una función de rotación. Entonces, para cada píxel de la imagen "ideal" se busca el píxel que le corresponde en la imagen rotada para tomar su valor de gris.

Ahora bien, de la expresión 4.13 se obtienen valores reales para x_i e y_i, mientras que las posiciones de los píxeles en la matriz son valores enteros. La aproximación más sencilla para calcular el nivel de gris de cada píxel de la nueva imagen, a partir de la formula 4.13, consiste en asignar el valor de gris del píxel de la imagen original cuya posición sea el resultado de truncar los valores de $\hat{x}_i$ e $\hat{y}_i$.

Para evitar el *aliasing* introducido al calcular el valor de gris de esta manera se suele suavizar, calculando el valor de gris del nuevo píxel, como la cantidad de gris que correspondería dependiendo del porcentaje de solapamiento con los cuatro vecinos del píxel, en caso que se tomase x_i e y_i como valores reales (ver ejemplo de la figura 4.9). Esto es, se hace una interpolación de Lagrange.

La mayoría de las técnicas de estimación del desencuadre pueden ser clasificadas dentro de los siguientes tipos generales dependiendo de la aproximación seguida [CC98, Hul98, HB04]: basados en proyecciones horizontales [Bai87, LFK02], basados en la transformada de Hough [AF00, HFD90, YJ96] y *clustering* de los vecinos más cercanos [JBWK99, LFK01, LT03].

Basada en proyecciones horizontales

Este tipo de aproximaciones parten de la suposición de que los documentos tienen el texto dispuesto a lo largo de líneas paralelas, cosa que ocurre en la gran mayoría de los documentos. Estos métodos son considerados los más rápidos y son simples de implementar.

La **proyección horizontal** consiste en sumar el nivel de gris de todas las columnas para cada fila (ecuación 4.14).

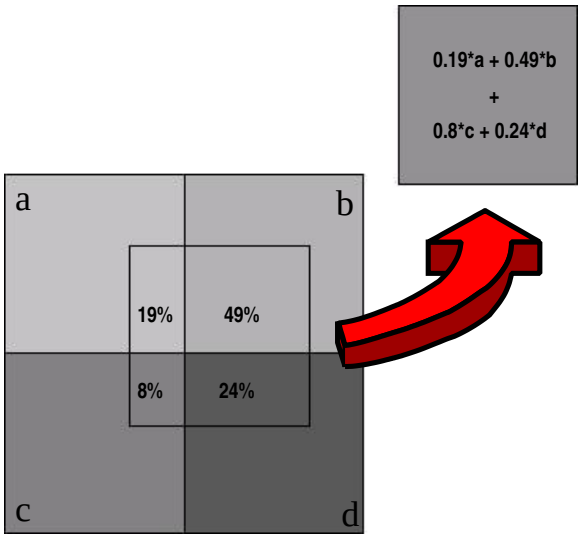


Figura 4.9: Ejemplo de suavizado de Lagrange. El píxel a rotar se obtiene en valores reales, con lo que no coincide con ningún píxel entero en la imagen original, sino que suele solaparse entre varios. El valor de gris del píxel resultante se calcula como el porcentaje de solapamiento con los píxeles a,b,c y d en la imagen original.

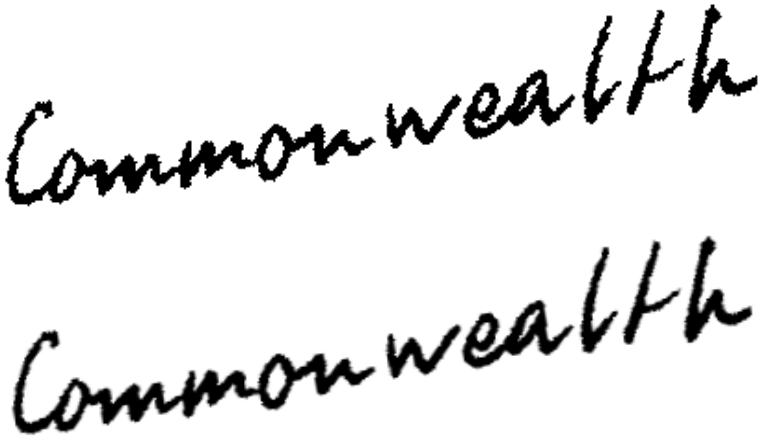


Figura 4.10: Ejemplo de rotación sin suavizado, figura superior y con suavizado mediante interpolación de Lagrange, figura inferior.

$$v(y) = \sum_{x} f(x, y) \tag{4.14}$$

En la imagen 4.11 se muestra la proyección horizontal de un párrafo de texto para diferentes ángulos de rotación. Como se aprecia en las proyecciones, cuanto menos desencuadre presenta una párrafo, su proyección presenta picos más grandes y valles más profundos. La proyección para la imagen sin *skew* presenta mayor amplitud y frecuencia. El problema de calcular el ángulo de desencuadre se reduce a un problema de optimización donde a partir de un conjunto de imágenes rotadas artificiálmente un rango de ángulos, encontrar aquella rotación que presente una la proyección horizontal cuya variación entre picos y valles sea mayor. Ahora sólo se necesita una función objetivo que puntúe la variación entre picos y valles para cada proyección. En esta tesis se utilizarán dos funciones objetivo. La primera función será la desviación típica de la proyección horizontal (ecuación 4.15) donde $filas$ es el número de filas de la proyección y μ es la media de la proyección. Para simplificar la notación, sea v_α la proyección horizontal para una imagen rotada artificialmente un ángulo α.

$$f(v_\alpha) = \sqrt{\sum_{1 \leq m \leq filas} \frac{(\mu - v(m))^2}{filas}} \tag{4.15}$$

La segunda función será la utilizada por Baird en [Bai87]. Baird reduce el número de puntos a proyectar eligiendo los puntos centrales de las componentes conexas. Si el documento se ha adquirido en niveles de gris, hay que umbralizarlo previamente. Las componentes conexas muy grandes, con respecto a la media de las componentes, se supone que pertenecen a bordes negros, o a imágenes, y se descartan. Baird no descarta las componentes muy pequeñas, pues según él, el ruido no afecta sustancialmente a su algoritmo. En la implementación realizada del algoritmo de Baird, componentes con menos de 3 píxeles se descartan. La función objetivo utilizada es la siguiente:

$$f(v_\alpha) = \sum_{1 \leq m \leq filas} v(m)^2 \tag{4.16}$$

Basada en la transformada de Hough

Estos métodos utilizan la transformada estándar de Hough (ecuación 4.17) que traslada las coordenadas cartesianas (x, y) de cada píxel negro al dominio polar (ρ, θ).

$$\rho = x\, cos\theta + y\, sin\theta \tag{4.17}$$

Todos los puntos (x, y) en el espacio cartesiano que estén alineados a lo largo de una recta, pasarán por el mismo punto (ρ, θ) en el dominio sinusoidal [DH74].

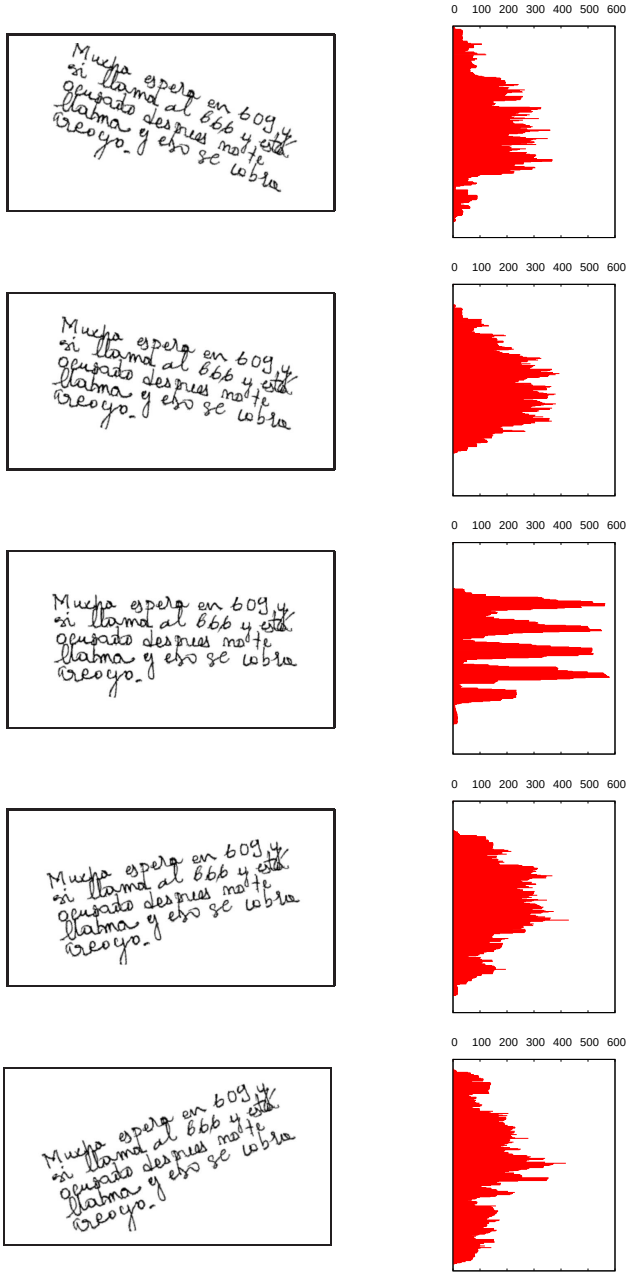


Figura 4.11: Párrafo de texto rotado con diferentes ángulos, columna izquierda y sus correspondientes proyecciones horizontales, columna derecha.

Así que se ha de contar todas las veces que se pasa por cada punto (ρ, θ), ya que representan puntos en el dominio cartesiano que están alienados a lo largo de una recta. Los máximos locales en el acumulador de Hough $H(\rho, \theta)$ representan lineas rectas, o lo que es lo mismo, representa las lineas rectas para las cuales tenemos más puntos en el dominio cartesiano. El máximo absoluto en $H(\rho, \theta)$ corresponderá a la linea con mayor número de puntos en la imagen original.

Algoritmo 2 Detección del ángulo de desencuadre (*skew*) basado en la transformada de Hough

1: **Input:** $P = \{(x, y) : (x, y) \in I(x, y)\}$ (puntos seleccionados de la imagen).
2: **Output:** $\hat{\theta}$ (ángulo de desencuadre).
3: **for all** $(x, y) \in P$ **do**
4: **for** $\theta_0 \leq \theta \leq \theta_n$ **do**
5: $\rho \leftarrow x\, cos\theta + y\, sin\theta$
6: $H(\rho, \theta) + +$
7: **end for**
8: **end for**
9: **return** $\hat{\theta} = \underset{(\rho,\theta)}{\text{argmax}}(H(\rho, \theta))$

Los métodos basados en la transformada de Hough tienen el inconveniente de que son lentos debido a que deben aplicar la transformada de Hough a todos los píxeles negros y para todo el rango de ángulos a detectar. El coste computacional de este método es $O(\frac{\Delta\theta}{\delta\theta} \cdot |P|)$. Para reducir el coste, la mayoría de autores intentan reducir $|P|$ [AF00] y/o el rango de ángulos $\Delta\theta$ (para la mayoría de aplicaciones con un rango entre $[-45 : 45]$ basta) y/o el incremento o paso de ángulo $\delta\theta$.

En [YJ96] se propone una utilización jerárquica de la transformada de Hough. Primero se hace un barrido del rango de ángulos, $\Delta\theta$, con un incremento de ángulo, $\delta\theta$, grande, con el fin de aproximar en pocos pasos el ángulo de desencuadre, y una vez se tiene el ángulo acotado, realizar una nueva exploración con un paso de ángulo más pequeño.

Para reducir el número de puntos a procesar $|P|$, después de umbralizar la imagen, se determinan las componentes conexas (8-conectadas) que hay en la imagen. Dependiendo del tamaño de las componentes, podemos clasificarlas en tres tipos: ruido, grandes y pequeñas. Si las componentes están formadas por un número reducido de píxeles (en nuestro caso menor o igual a 5) se descartan. De esta manera, el ruido tipo *pimienta* se elimina. La transformada de Hough es muy sensible a este tipo de ruido. Las componentes demasiado grandes (3 veces la media suele ser suficiente) que suelen corresponderse con imágenes, o bordes negros también se descartan, esto permite trabajar con documentos genéricos con una complejidad de composición alta. Los córporas utilizados en este trabajo no incluyen imágenes ni bordes negros, por lo que no hay componentes demasiado grandes.

Para el presente trabajo se han probado tres conjuntos de puntos para ser utilizados en la transformada de Hough: todos los píxeles negros de la imagen, los

puntos centrales de las cajas de inclusión y los centroides de cada componente conexa.

Basada en *clustering* de los vecinos más cercanos

Estos métodos suelen empezar con un proceso de etiquetado de componentes conexas de la imagen. Se requieren imágenes binarias, por lo que si la imagen fue adquirida con niveles de gris, hay que umbralizarla previamente. Para evitar ruido, y eliminar figuras y bordes negros, todas las componentes que sean mucho más grande, o mucho más pequeñas que la media de componentes, se eliminan. A partir de aquí, se van agrupando bloques con características similares formando componentes más grandes. Finalmente se intenta estimar el ángulo de desencuadre para estas componentes. Hay que decir que cuanto mayor número d componentes conexas mayor es la precisión del método.

En [HYR86], para cada componente conexa se localiza la componente más cercana y se calcula el ángulo de la recta que pasa por sus centroides. Los ángulos de cada par de componentes se acumulan en un histograma. El ángulo con mayor frecuencia del histograma es tomado como ángulo de *skew*. El problema de este método es su sensibilidad al ruido y a pequeñas variaciones posicionales. Este problema se produce debido a que el cálculo del ángulo suele hacerse para pequeñas distancias.

Jiang et al. [JBWK99] proponen un método (denominado por ellos *Focused Nearest-Neighbor Clustering* o FNNC) para elegir entre el conjunto de k vecinos más cercanos, $\{Q_1, \cdots, Q_k\}$ a cada componente conexa, P, un par (Q_i, Q_j) cuya distancia en perpendicular a P sea menor. El par, $\{(P, Q_i), (P, Q_j), (Q_i, Q_j)\}$ con la distancia mayor entre ellos se toman como una primera aproximación. A partir de la recta que pasa por esos dos puntos, se seleccionan todos los puntos que estén entre ella en un intervalo en perpendicular de distancia D. Mediante la técnica de los mínimos cuadrados se ajusta una recta a los puntos seleccionados, cuya pendiente se toma como ángulo de skew para el punto P, y se acumula en un histograma. El ángulo con mayor frecuencia se toma como ángulo de *skew*.

Lu et al. [LT03] resuelven el problema de las distancias demasiado cortas entre componentes utilizando lo que ellos llaman cadenas de vecinos más cercanos (*Nearest-Neighbor Chain*, o NNC). Definen unas reglas para elegir el vecino más cercano para cada componente conexa, entre las que cabe destacar que la altura de las cajas de inclusión mínima tengan una altura parecida, que la $\Delta x > \Delta y$ o que el vecino más próximo ha de estar a la izquierda de la componente actual. Luego intenta formar cadenas de vecinos de k componentes $[C_1, \cdots, C_k]$ para los que C_{i+1} es el vecino más próximo de C_i para $i = 1, 2 \cdots, k-1$. Si el número de k-NNC es menor que un umbral dado, se repite el proceso para $k-1$. El ángulo de *skew* se calcula en base a los centroides de las componentes C_1 y C_k.

En este trabajo, para cada componente conexa (con un mínimo de píxeles), se busca el vecino más próximo a su caja de inclusión. El ángulo de la recta que pasa por los dos puntos centrales de dichas cajas de inclusión se acumula en un

histograma de ángulos. El máximo del histograma será tomado como ángulo de *skew*.

4.1.4. Segmentación en lineas

La segmentación de una página de texto requiere un análisis del su contenido. Los sistemas de análisis de página intentan obtener una representación jerárquica de la misma, donde cada bloque representa una zona homogenea de la página: una imagen, una columna, cabecera de texto, etc. La mayoría de los sistemas de análisis de página se pueden clasificar en: basados en componentes conexas [SPJ97, WY95, Zla94]; basados en RLSA [TA92] o basados en proyecciones [KNSV93].

Una vez analizada la página, los bloques de texto necesitan ser descompuestos en lineas de texto. En este trabajo solamente se abordará el problema de la segmentación de bloques de texto en lineas de texto. No se pretende dividir el bloque en frases, ya que para ello habría que analizar sintácticamente la frase cosa que no puede hacerse sin saber lo que hay escrito.

El método utilizado para segmentar los bloque de texto está basado en proyecciones horizontales. Los valles de estas proyecciones son posibles puntos de segmentación. Si un valle tiene un valor de cero, representa una zona de la imagen que no contienen texto, y por esa razón es descartada y sus fronteras son tomadas como candidatos a puntos de segmentación. Si los ascendentes y los descendentes de dos lineas se cruzan o tocan, provocan valles con valores superiores a cero. Se utiliza un umbral ρ a partir del cual los valles delimitarán las zonas de texto.

La utilización previa del algoritmo RLSA sobre la imagen produce un efecto derivativo sobre la proyección, exagerando más la distancia entre valles y picos (ver imagen 4.12).

4.2. Normalización a nivel de texto

Las técnicas expuestas en este apartado pretenden normalizar el estilo de escritura, con lo cual se podría decir que son técnicas locales.

4.2.1. Corrección del *slope*

El *slope* es la pendiente o inclinación que presenta la linea base sobre la que esta escrita una palabra, o una secuencia de ellas (fig. 4.13), con respecto al eje de ordenadas. Una vez determinado el ángulo, la corrección consiste en aplicar una rotación de la imagen con el mismo ángulo que el de *slope* en sentido contrario.

El primer paso para determinar el ángulo de *slope* es dividir cada frase en segmentos de frase. Este proceso no pretende segmentar el texto en palabras, sino dividirlo en aquellas secuencias de texto que estén muy cercanas entre sí. Se asume que si están cercanas entre sí fueron escritas como una unidad, compartiendo todas las características del estilo de escritura, entre las que se en cuenta el ángulo de *slope*. Esta división suele basarse en heurísticos más o menos sofisticados. Se puede

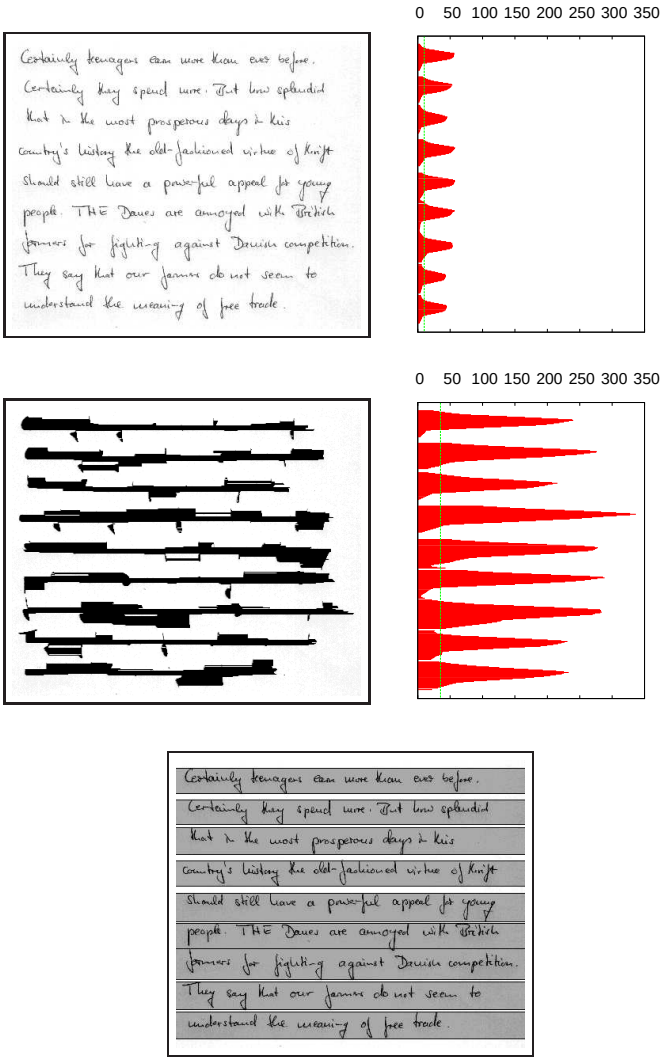


Figura 4.12: Figuras superiores, ejemplo de proyección horizontal para un bloque de texto. Figuras centrales, ejemplo del efecto derivativo que produce el aplicar previamente el algoritmo RLSA. En la parte inferior, resultado de la segmentación, las partes sombreadas corresponden con los distintos segmentos.

Figura 4.13: Ejemplo de palabra con *slope* (β).

fijar un tamaño mínimo de espacio en blanco que sirve para delimitar los segmentos de palabras. En este trabajo, en la detección de huecos se realiza una proyección horizontal de la imagen, se obtiene el tamaño medio de los huecos encontrados, tomando como hueco todo aquel espacio de la proyección para el cual el nivel de gris esté por debajo de un umbral determinado. Este tamaño medio más una constante empírica determinará los puntos de corte entre tramos.

En este trabajo se estudian dos técnicas de corrección del *slope*, la primera basada en ajuste de líneas a los perfiles y otra en proyecciones horizontales, similar a la utilizada para la estimación del ángulo de *skew* en la sección 4.1.3.

Basada en ajuste de línea a los perfiles

La determinación del ángulo se realiza en tres pasos:

1) *Emborronado* de cada segmento de frase original con el algoritmo de suavizado por longitud de píxeles horizontales consecutivos RLSA [WCW82] (algoritmo 1. Previamente se binariza la imagen con algún método global como por ejemplo Otsu.

2) De la imagen obtenida del paso anterior, se calcula para cada segmento el contorno superior C_s y el inferior C_i. Usualmente se busca el primer y el último píxel negro de cada columna. Las partes horizontales de los ascendentes, como por ejemplo el trazo horizontal de la *t*, puede causar que el algoritmo devuelva un contorno erróneo (ver figura 4.14). Con una sencilla variación del algoritmo se pueden evitar estos problemas. La variación consiste en realizar un RLSA tanto horizontal como vertical, para posteriormente localizar por columna el tramo de píxeles de imagen consecutivos de mayor longitud. Los píxeles superior e inferior de este tramo se tomarán como fronteras para dicha columna.

3) El último paso consiste en ajustar una recta a cada uno de los contornos (C_s y C_i). Primero se eliminan los puntos anómalos. Para cada una de las fronteras se obtiene la media $\overline{y}$, y la desviación típica σ, de sus y's. Todos aquellos puntos cuyas y's estén entre el valor medio más menos la desviación típica

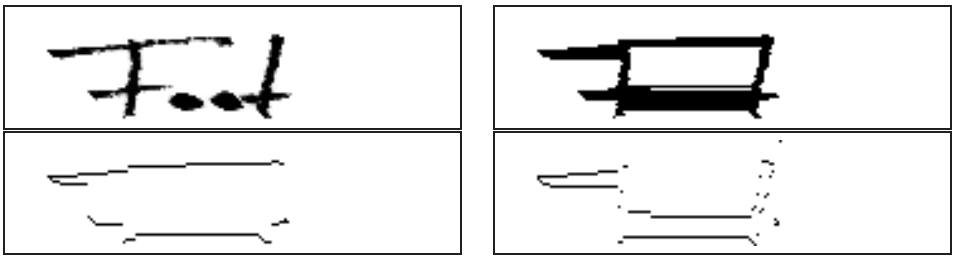


Figura 4.14: Ejemplo de problema en la obtención de los contornos. De izquierda a derecha y de arriba a abajo: imagen original; resultado de aplicar sobre la imagen original el algoritmo RLSA; contorno erróneo; contorno correcto.

son tomados en cuenta para el ajuste de las líneas $\hat{C}_{\{s,i\}} = \{(x,y) : y \leq |\overline{y} - \sigma|\}$. Las rectas se pueden ajustar de varias maneras, por ejemplo, ajuste por vector propio o por mínimos cuadrados [DH74]. Del promedio de ambas rectas se obtiene una tercera, cuya pendiente será tomada como ángulo del *slope*.

El último paso consiste en rotar el tramo un ángulo igual al ángulo del *slope* en sentido contrario, con lo cual se obtiene un texto alineado horizontalmente. En la figura 4.15 se muestra un ejemplo de los pasos seguidos para corregir el *slope* mediante esta técnica.

Basada en proyecciones horizontales

En la imagen 4.16 se muestra la proyección horizontal (ver sección 4.1.3) para la palabra *commonwealth* para diferentes ángulos de rotación. Como se aprecia en las proyecciones, cuanto menos *slope* presenta una palabra, su proyección presenta picos más grandes. El problema de calcular el ángulo de *slope* se reduce a un problema de optimización donde a partir de un conjunto de imágenes rotadas artificiálmente un rango de ángulos se ha de encontrar aquella para la cual su proyección horizontal presente el pico mayor. Ahora sólo se necesita una función objetivo que puntúe el tamaño máximo de cada proyección horizontal. En [KDFK03, Kavalleratou03] se utiliza como función objetivo la función de energía de Wigner-Ville, en otros trabajos se utiliza la varianza como en [KB06]. En esta tesis se utilizara como función objetivo la función máximo de la proyección horizontal (ecuación 4.18) y la desviación típica [PTV04] (ecuación 4.19). Para simplificar la notación, υ_α será la proyección horizontal para un segmento de frase que ha sido rotado previamente un ángulo α.

$$f(\upsilon_\alpha) = \max_{0 \leqslant i < cols} \upsilon_\alpha(i) \tag{4.18}$$

donde $\upsilon_\alpha(i)$ corresponde a la columna i de la proyección horizontal υ_α, y $cols$ es el número de columnas de la imagen. En la ecuación 4.18 se tomará como ángulo

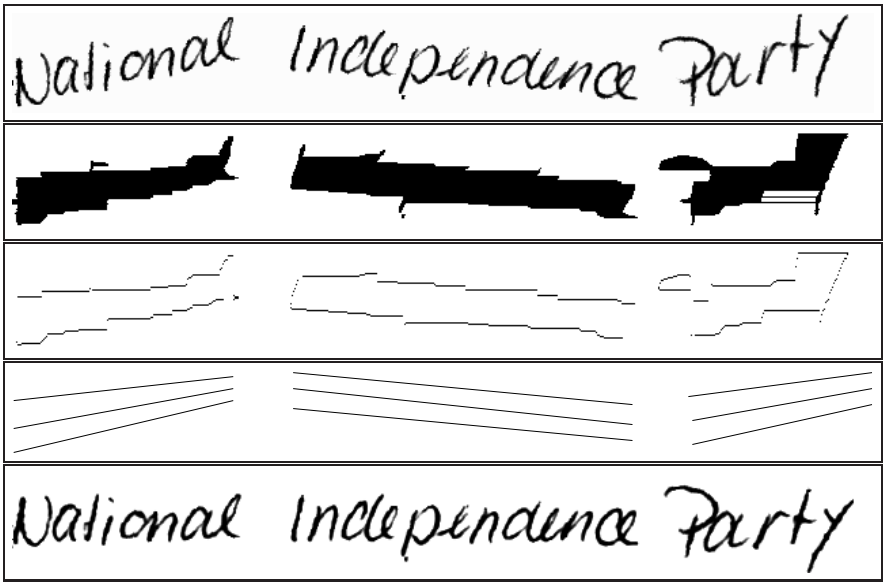


Figura 4.15: Ejemplo de corrección del *slope* basado en ajuste de línea a los perfiles. De arriba a abajo: texto con slope; texto suavizado con el algoritmo RLSA para los tres tramos detectados en la imagen; contornos superior e inferior de la imagen anterior; Líneas ajustadas al contorno superior e inferior y línea promedio; texto con el *slope* corregido.

de *slope* el de la proyección horizontal, en sentido contrario, que presente el pico mayor.

$$f(v_\alpha) = \sqrt{\sum_{1 \leq m \leq cols} \frac{(\mu - v_\alpha(m))^2}{cols}} \tag{4.19}$$

En la ecuación 4.19, $cols$ es el número de columnas de la proyección horizontal, mientras que μ es la media de la proyección por columna.

Los métodos basados en proyecciones horizontales encuentran dificultades para tratar con palabras, o segmentos de palabra cortos, ya que no tiene porque existir una notable diferencia entre el alto y el ancho. Así que para las diferentes proyecciones tampoco existe una diferencia importante que sirva para una correcta discriminación. Para evitar este efecto no deseado, cuando se segmenta en segmentos de frase se fuerza a que haya una longitud mínima.

Experimentación

En la tabla 4.1 se presentan los resultados obtenidos para los corpus ODEC y IAMDB dependiendo del método de corrección usado. El método de corrección del ángulo de *slant* utilizado en esta experimentación es el de *detección de bordes*,

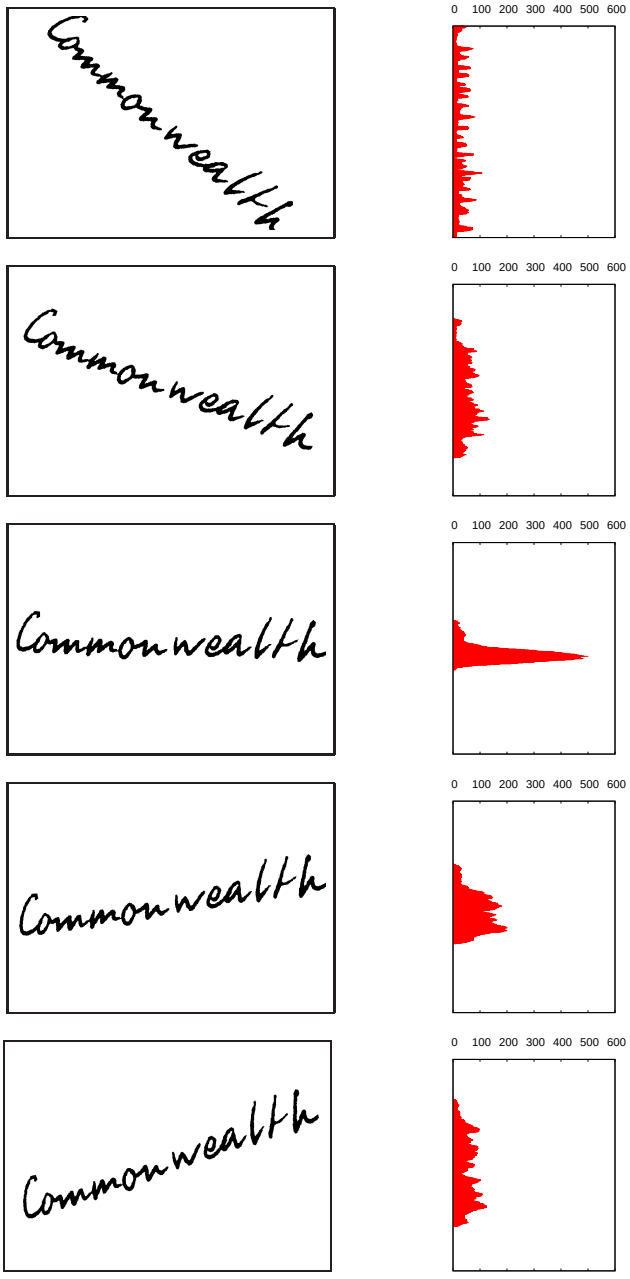


Figura 4.16: La palabra *commonwealth* rotada con diferentes ángulos. En la columna derecha se pueden ver las respectivas proyecciones horizontales.

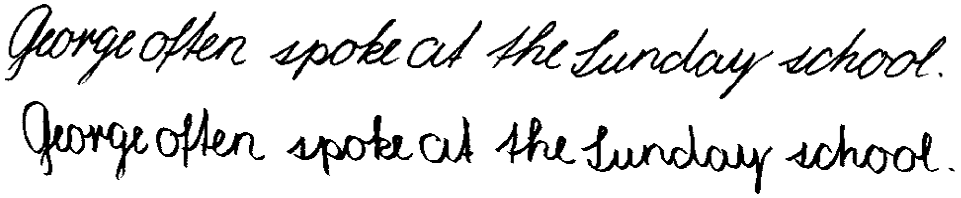


Figura 4.17: Ejemplo de texto con *slant*. En la imagen superior se puede apreciar que los caracteres presentan una desviación en sus componentes verticales con respecto al eje vertical. En la imagen inferior se muestra la misma imagen con el *slant* corregido.

mientras que el método de normalización del tamaño de los caracteres es el de *escalado de ascendentes y descendentes*.

método	ODEC	IAMDB
baseline	28.6	39.3
Ajuste a perfiles	**27.1**	29.9
Máximo	**27.1**	**28.6**
Desviación típica	27.9	31.4

Tabla 4.1: Tabla comparativa para los distintos métodos de estimación del ángulo de *slope* y para los corpus ODEC y IAMDB. Los valores representados corresponden a los valores WER (ver sección 2.3.3). En el panel superior de la tabla: resultados de reconocimiento sin corrección del *slope*. En el panel central: resultados para el métodos de ajuste de linea a los contornos. En el panel inferior métodos basados en proyecciones horizontales: función de optimización máximo y desviación típica.

Para el corpus ODEC, el mejor resultado obtenido es de 27.1 WER con la técnica de *ajuste a perfiles* o con la de *proyecciónes horizontales* con función de optimización *máximo*. La mejora relativa respecto a no corregir el *slope* es del 5.2 %. Para el corpus IAMDB el mejor resultado obtenido es de 28.6 WER con la técnica de *proyecciónes horizontales* con función de optimización *máximo*. La relativa, en este caso con respecto a no realizar corrección de *slope* es del 23.9 %. Lo primero que cabe destacar es la importancia de realizar la corrección del *slope*, mientras que la diferencia entre los distintos métodos no resulta muy significativa.

4.2.2. Corrección del *slant*

Una de las características más importantes que define un estilo de escritura es el ángulo, en sentido horario, que presentan las componentes verticales de los caracteres respecto al eje vertical. Esta inclinación es conocida como *slant* (fig. 4.17).

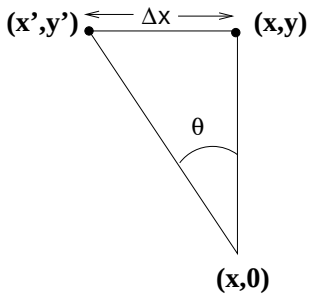


Figura 4.18: Función *shear*: el píxel (x, y) es desplazado a la posición (x', y') dependiendo de su altura (y) y del ángulo (θ).

La corrección del *slant* se realiza en dos fases, primero se estima el ángulo de *slant* del texto, y en un segundo paso se corrige aplicando una operación *shear* (ver fig. 4.18) con el ángulo inverso al de *slant*. Supongamos que α es el ángulo de *slant* de un determinado texto, *shear* es una función que resitúa cada punto de la imagen original de la siguiente manera:

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = shear \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_i + y_i\, tan(-\alpha) \\ y_i \end{pmatrix} \tag{4.20}$$

Para estimar el ángulo de slant se han propuesto diversos métodos. En este trabajo se han estudiado aproximaciones basadas en métodos estructurales, en detección de bordes y en proyecciones verticales. Estos métodos se exponen a continuación.

Métodos estructurales

Los métodos estructurales, después de adelgazar el texto (*thinning*), lo codifican utilizando código 8-direccional (*chain codes*) que aproximan la dirección de cada tramo. Se seleccionan aquellos tramos cuyos ángulos estén cerca de la vertical. La media de los ángulos de dichos tramos, ponderados con su longitud, se asume como ángulo de *slant* [YK03, KGS99]. En [MB01], el contorno casi vertical se calcula teniendo en cuenta las transiciones blanco-negro y negro-blanco en sentido horizontal. Se obtiene el histograma de la distribución de los ángulos de los contornos casi verticales. En [SR98] el contorno se obtiene de la aplicación del filtro de detección de bordes de Canny.

Método basado en detección de bordes o *edges*

Edges o bordes son regiones de la imagen donde se encuentra un fuerte contraste en la intensidad. Los bordes son fácilmente detectables, pero con un alto coste computacional, en el dominio de la frecuencia, utilizando la transformada de

Fourier y un filtro paso-alto. Como los bordes corresponden a variaciones en el gradiente de la intensidad, también pueden ser calculados utilizando las derivadas de intensidad. La posición de los bordes puede ser obtenida buscando máximos en las derivadas primeras, o buscando pasos-por-cero en las segundas derivadas.

Las derivadas pueden ser aproximadas mediante la convolución de la imagen con un *kernel* apropiado. Convolución consiste en una suma ponderada del contexto para cada píxel de la imagen (ver sección 4.1.2).

Los detectores de bordes más usuales suelen estar basados en los *kernels* de Sobel. Típicamente estos métodos [YS98, SS97] hacen una convolución de la imagen utilizando los *kernels* vertical y horizontal de Sobel (matrices de la figura 4.21), obteniendo así dos imágenes convolucionadas, Gy y Gx.

$$K_{S_h} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \qquad K_{S_v} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{4.21}$$

El gradiente explica como está cambiando la intensidad luminosa en cada punto de la imagen. El ángulo fase del gradiente, $\theta_{i,j}$, explica en que dirección está cambiando, mientras que su módulo, $M_{i,j}$, da cuenta de como de rápido se está cambiando (ecuaciones 4.22). En el caso que atañe a este método, sólo interesa la dirección.

$$\theta_{i,j} = \arctan\left(\frac{Gy_{i,j}}{Gx_{i,j}}\right) \qquad M_{i,j} = \sqrt{Gx_{i,j}^2 + Gy_{i,j}^2} \tag{4.22}$$

El ángulo de fase del gradiente se calcula para cada punto de la imagen. Seguidamente se genera el histograma de frecuencias de ángulos de fase mediante el recuento del número de apariciones de cada magnitud de ángulo en θ. Con el objetivo de restar importancia a los ángulos que estén más alejados de la vertical respecto aquellos que están más cercanos a ella, se hace necesario aplicar un suavizado a dicho histograma. Para ello se suele aplicar un filtro triángulo unidad, $F_t(\alpha)$, centrado en 90 grados o un filtro gausiano $F_g(\alpha)$ (ver el ejemplo de la figura 4.19).

$$F_t(\alpha) = \left(1 - \frac{|90 - \alpha|}{90}\right) \qquad F_g(\alpha) = \left(\frac{1}{\sqrt{2\pi}\sigma}\, e^{-\frac{1}{2}\left(\frac{(\alpha-\mu)}{\sigma}\right)^2}\right) \tag{4.23}$$

Finalmente se computa la media, que es tomada como valor del ángulo de *slant* dominante del texto. Una vez determinado el ángulo medio, la corrección del *slant* consiste en aplicar la función *shear* para dicho ángulo cambiado de signo, a los píxeles de la imagen tal como se muestra en la figura 4.18.

Métodos basado en proyecciones verticales

Proyección vertical: consiste en sumar el nivel de gris de todas las filas de cada columna (ecuación 4.24).

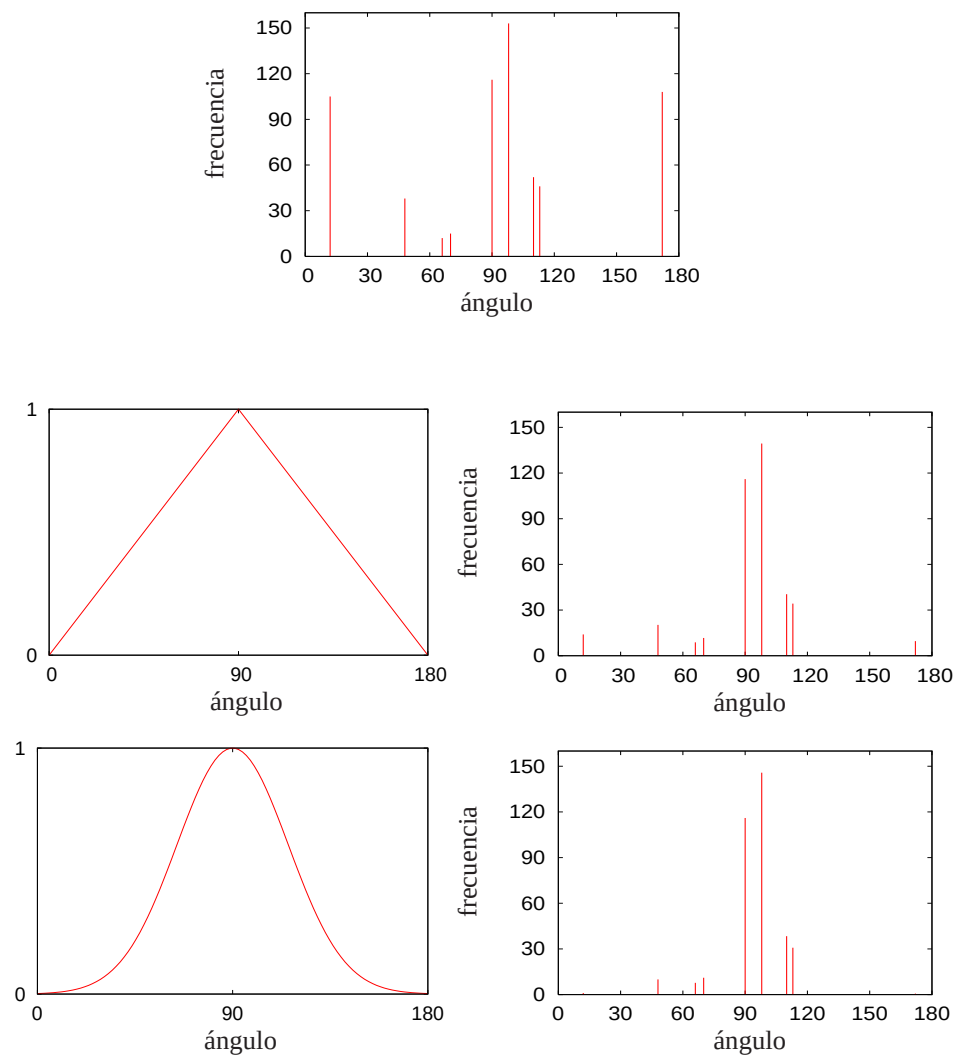


Figura 4.19: De arriba a bajo y de izquierda a derecha: histograma de ángulos de fase, filtro triángulo unitario, histograma suavizado con filtro triángulo, filtro gausiano, resultado de filtrar el histograma suavizado con filtro gausiano.

Algoritmo 3 Corrección del Slant: método Sobel

input: I_{entr}
output: I_{sal}, α_{slant}
1: $G_y \leftarrow K_{S_h} \otimes I_{ent}$
2: $G_x \leftarrow K_{S_v} \otimes I_{ent}$
3: **for all** (x,y) $\in I_{ent}$ **do**
4: $\quad \theta(x,y) \leftarrow arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right)$
5: **end for**
6: $Hist(\alpha) \leftarrow frec(\alpha = \theta(x,y)) \quad \alpha \in [-90, 90] \forall x, y$
7: $Hist_s(\alpha) \leftarrow Hist(\alpha) F_s(\alpha) \quad \alpha \in [-90, 90]$
8: **return** $\overline{Hist_s}(\alpha)$

$$\upsilon(x) = \sum_y f(x,y) \tag{4.24}$$

Los métodos basados en proyecciones verticales parten de la observación de que la proyección vertical para una imagen sin *slant* presenta la máxima variabilidad entre picos y valles (ver ejemplo de figura 4.20). Estos métodos aplican una operación *shear* (ver figura 4.18), sobre la imagen original, para un conjunto discreto de ángulos, obteniendo así un conjunto de imágenes con un *slant* artificial. Suponiendo que el texto presenta un ángulo de *slant*, α, respecto a la vertical, al aplicarle una operación *shear* con un ángulo $-\alpha$, producirá un texto sin *slant*. Para cada imagen se obtiene su proyección vertical. Ahora el problema de encontrar el ángulo de *slant* se puede expresar como el problema de encontrar la proyección vertical que presente la máxima variabilidad. Así pues, los métodos basados en proyecciones se pueden unificar y ser expresados como problemas de optimización donde se proponen diferentes funciones objetivo:

$$\hat{\alpha} = \underset{\alpha \in [45:135]}{\text{argmax}} \; f(\upsilon_\alpha) \tag{4.25}$$

Para simplificar la notación diremos que υ_α es la proyección vertical para una imagen a la que previamente se le ha aplicado una operación *shear* con un ángulo α. Parece de sentido común que el rango de ángulos de *slant* se pueda acotar entre -45 y 45 grados con respecto a la vertical, $\alpha \in [45, 135]$. A la hora de elegir funciones objetivo que nos estimen la variabilidad entre picos y valles aparecen diferentes alternativas. A continuación se exponen algunas de ellas.

Función objetivo: Método IDIAP

Esta función objetivo fue introducida por Vinciarelli [VL00, VL03] investigador del instituto IDIAP[1]. En ella se tienen en cuenta aquellas columnas de la

[1] http://www.idiap.ch

proyección vertical que pertenezcan estrictamente a trazos continuos verticales. El criterio de selección de columnas de la proyección pasa por crear una proyección normalizada donde se muestre el porcentaje de trazo continuo dentro de cada columna. Para ello, después de umbralizar la imagen, cada columna de la proyección vertical se dividirá por la distancia entre el primer y el último píxel negro en la correspondiente columna de la imagen original. Como el numerador será menor o igual que el denominador, el rango del valor de cada columna estará entre 0 y 1. Si en la imagen una columna contiene un trazo continuo, este será de la misma longitud que su proyección vertical. Hay que tener en cuenta que una proyección vertical es equivalente a sacar de cada columna todas las filas que contengan blanco. La proyección normalizada se calcula del siguiente modo:

$$C_\alpha(m) = \frac{\upsilon_\alpha(m)}{\triangle y_\alpha(m)} \tag{4.26}$$

donde $\triangle y_\alpha(m)$ es la distancia entre el primer y último píxel de la columna m en la imagen original. Si la columna m de la imagen contiene un trazo continuo, sin ningún blanco intercalado, entonces $C_\alpha(m) = 1$, si no, $C_\alpha(m) \in [0, 1[$. Para medir la variabilidad de una proyección, se eligen aquellas columnas de la proyección vertical para las cuales su proyección normalizada vale 1. La función objetivo será pues:

$$f(\upsilon_\alpha) = \sum_{\{m:C_\alpha(m)=1\}} \upsilon_\alpha(m)^2 \tag{4.27}$$

Esta función objetivo presenta ciertas dificultades con estilos de escritura que utilicen caracteres con formas redondeadas, o con caracteres como *i* o la *j* que tienen punto sobre ellos. También es muy sensible a ruido del tipo *sal y pimienta*. Restringir el calculo a columnas que estrictamente pertenezcan a trazos verticales puede ser una restricción demasiado dura. Se necesita introducir algún tipo de suavizado a la función. En lugar de tener en cuenta las columnas de la proyección que estrictamente pertenezcan a trazos verticales continuos, aquellas para las cuales se cumpla la restricción $C_\alpha(m) = 1$, se utilizarán aquellas columnas que contengan un trazado vertical cuasi-continuo, $C_\alpha \geq \rho$ donde $\rho \in [0, 1]$. La función objetivo resulta como sigue:

$$f(\upsilon_\alpha, \rho) = \sum_{\{m:C_\alpha(m)\geq\rho;\rho\in[0,1]\}} \upsilon_\alpha(m)^2 \tag{4.28}$$

Función objetivo: Desviación típica

En esta aproximación, se utiliza como función objetivo para medir la variabilidad de cada proyección vertical un estadístico bien conocido como es la desviación típica [PTV04, PTRV06]

Algoritmo 4 Corrección del Slant: Función objetivo Idiap.

input: $I_{ent}(x, y)$
output: α

$Proj(\alpha, x) \leftarrow 0 \quad \forall\alpha \in [-45, 45] \wedge \forall x$
for all y **do**
  for all $\alpha \in [-45, 45]$ **do**
    $Col\_Disp(\alpha) \leftarrow y \cdot tan(\alpha)$
  end for
  for all x **do**
    if $(y, x) \in I_{ent} =$ BLACK **then**
      for all $\alpha \in [-45, 45]$ **do**
        $Proj(\alpha, x + Col\_Disp(\alpha)) + +$
      end for
    end if
  end for
end for
for all x **do**
  $y_u \leftarrow 1$
  while $(x, y_u)!$ =BLACK **do**
    $y_u + +$
  end while
  $y_d \leftarrow rows$
  while $(x, y_d)!$ =BLACK **do**
    $y_d - -$
  end while
  $\Delta Y \leftarrow y_d - y_u$
end for
for all $\alpha \in [-45, 45]$ **do**
  for all x **do**
    $C(\alpha, x) \leftarrow Proj(\alpha, x)/\Delta Y$
  end for
end for
for all $\alpha \in [-45, 45]$ **do**
  for all x **do**
    if $C(\alpha, x) == 1$ **then**
      $f(\alpha) \leftarrow f(\alpha) + (Proj(\alpha, x))^2$
    end if
  end for
end for
return $\underset{\alpha\in[-45,45]}{\text{argmax}}\ (f(\alpha))$

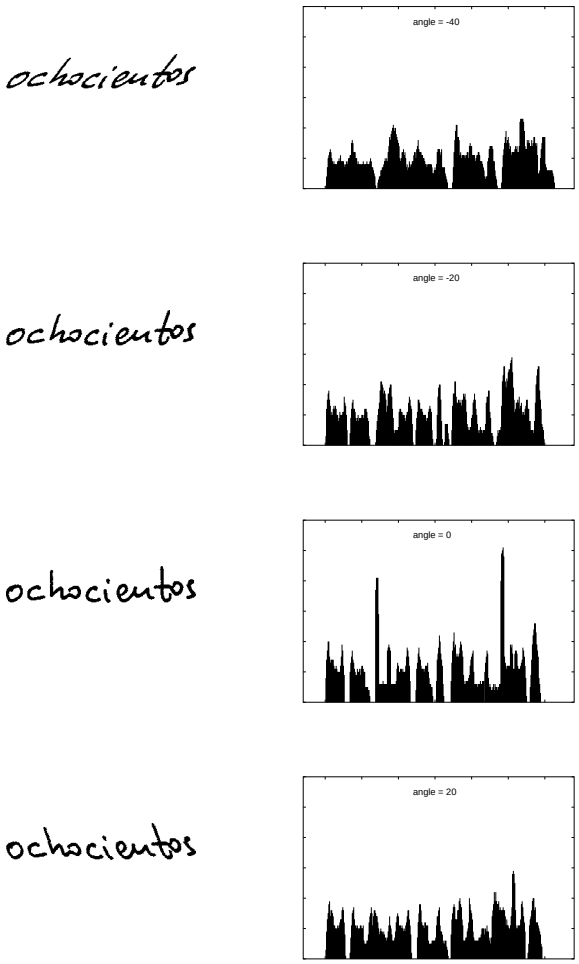
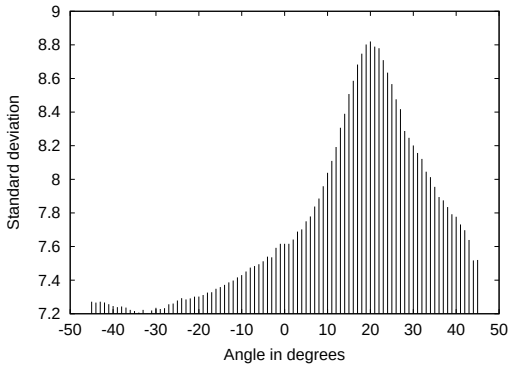


Figura 4.20: En el ejemplo se aprecia que cuanto menor *slant* tiene una palabra, más amplitud y frecuencia tiene su proyección.



$$f(v_\alpha) = \sqrt{\sum_{1 \leq m \leq cols} \frac{(\mu - v_\alpha(m))^2}{cols}} \tag{4.29}$$

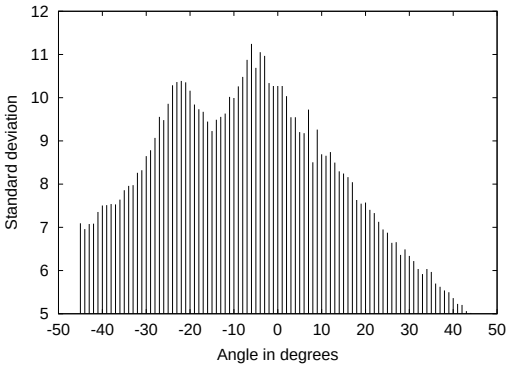
donde $cols$ es el número de columnas de la proyección vertical, y μ es la media de la proyección vertical por columna.

Algoritmo 5 Corrección del Slant: función objetivo Máxima Varianza.

input: $I_{ent}(x, y)$
output: α

$Proj(\alpha, x) \leftarrow 0 \quad \forall \alpha \in [-45, 45] \wedge \forall x$
for all y **do**
  for all $\alpha \in [-45, 45]$ **do**
    $Col\_Disp(\alpha) \leftarrow y \cdot tan(\alpha)$
  end for
  for all x **do**
    if $(y, x) \in I_{ent} =$ BLACK **then**
      for all $\alpha \in [-45, 45]$ **do**
        $Proj(\alpha, x + Col\_Disp(\alpha)) + +$
      end for
    end if
  end for
end for
for all $\alpha \in [-45, 45]$ **do**
  $StdDev(\alpha) \leftarrow GetStadDev(Proj, \alpha)$
end for
return $\underset{\alpha \in [-45,45]}{\text{argmax}} \ (StdDev(\alpha))$

Desafortunadamente, es bastante usual encontrar textos con más de un ángulo de *slant* dominantes. Esto suele ocurrir cuando hay grandes ascendentes y/o descendentes con diferente inclinación. En estos casos el ángulo dominante ($\hat{\alpha}$) puede



que no sea el ángulo más representativo para el *slant* total. Se hace necesario aplicar algún tipo de suavizado. El suavizado estudiado consiste en tener en cuenta aquellos ángulos para los cuales su desviación estándar esté bastante cerca del máximo. La medida de cercanía usada para medir es una fracción($\rho \in [0,1]$) de la mayor desviación estándar. El conjunto de los ángulos a tomar en cuenta es:

$$R_\rho = \{\alpha : f(v_\alpha) \geq \rho f(v_{\hat{\alpha}}); \rho \in [0,1]\} \tag{4.30}$$

El ángulo suavizado resultante $\bar{\alpha}$, se calcula como el centro de masa de los ángulos cuya desviación estándar sea mayor o igual que $\rho\ f(v_{\hat{\alpha}})$,

$$\bar{\alpha} = \frac{\sum\limits_{\alpha \in R_\rho} \alpha\ f(v_\alpha)}{\sum\limits_{\alpha \in R_\rho} f(v_\alpha)} \tag{4.31}$$

Función objetivo: Longitud del perfil

Esta función objetivo está basada en la longitud del contorno superior de la proyección vertical (ecuación 4.32). La idea es que a mayor variabilidad mayor longitud tendrá el contorno superior. Además, esta función tiene en cuenta no sólo la variabilidad de la proyección, sino que realza la importancia de la alternancia entre picos y valles.

$$f(v_\alpha) = \sum_{m=1}^{cols-1} \sqrt{1 + (v_\alpha(m) - v_\alpha(m+1))^2} \tag{4.32}$$

Para ahorrar computo, la ecuación 4.32 se puede aproximar como:

$$f(v_\alpha) = \sum_{m=1}^{cols-1} |v_\alpha(m) - v_\alpha(m+1))| \tag{4.33}$$

Algoritmo 6 Corrección del Slant: función objetivo longitud del perfil.

input: $I_{ent}(x, y)$
output: α

1: $Proj(\alpha, x) \leftarrow 0 \quad \forall\alpha \in [-45, 45] \wedge \forall x$
2: **for all** y **do**
3: **for all** $\alpha \in [-45, 45]$ **do**
4: $Col\_Disp(\alpha) \leftarrow y \cdot tan(\alpha)$
5: **end for**
6: **for all** x **do**
7: **if** $(y, x) \in I_{ent} =$ BLACK **then**
8: **for all** $\alpha \in [-45, 45]$ **do**
9: $Proj(\alpha, x + Col\_Disp(\alpha)) + +$
10: **end for**
11: **end if**
12: **end for**
13: **end for**
14: **for all** $\alpha \in [-45, 45]$ **do**
15: $long(\alpha) \leftarrow GetLong(Proj, \alpha)$
16: **end for**
17: **return** $\underset{\forall\alpha\in[-45,45]}{\text{argmax}} \; (long(\alpha))$

Función objetivo: Método Zimmermann

Zimmermann [Zim03] define como función objetivo la suma de todas las columnas de la proyección ponderada (ecuación 4.34).

$$f(\upsilon'_\alpha) = \sum_{m=1}^{cols-1} \upsilon'_\alpha(m) \tag{4.34}$$

A esta proyección ponderada υ'_α, el autor la denomina *proyección generalizada* y la define como:

$$\upsilon'_\alpha(x) = \sum_y p_{x,y} f(x, y) \tag{4.35}$$

donde:

$$P_{x,y} = \left\{ \begin{array}{lcl} 0 & ; & f(x, y) == BLANCO \;\; o \;\; y \leq 0 \\ p_{x,y-1} + 1 & ; & f(x, y) == NEGRO \end{array} \right\} \tag{4.36}$$

Esta proyección consiste en una suma ponderada de píxeles para cada columna (ecuación 4.35) . Una vez umbralizada la imagen, la ponderación (ecuación 4.36) consiste en puntuar cada píxel dependiendo de la cantidad de píxeles *foreground* previos a él. Si un píxel pertenece a la clase *background* su peso es cero, en caso contrario, es el peso del píxel anterior más uno. De esta manera se premia a

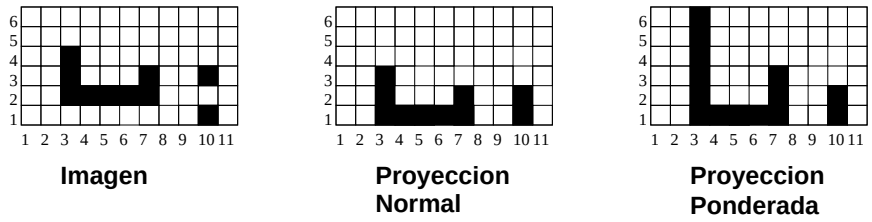


Figura 4.21: Ejemplo de proyección *normal* y de proyección ponderada. La ponderación para cada píxel de la columna tres sería: $\upsilon'_\alpha(3) = 0f_{3,6} + 0f_{3,5} + 1f_{3,4} + 2f_{3,3} + 3f_{3,2} + 0f_{3,1} = 6$. En cambio para la columna 10 sería: $\upsilon'_\alpha(10) = 0f_{10,6} + 0f_{10,5} + 0f_{10,4} + 1f_{10,3} + 0f_{10,2} + 1f_{10,1} = 2$, ya que sólo hay dos píxeles negros y no son consecutivos.

aquellos píxeles que pertenecen a trazos verticales (ver ejemplo de la figura 4.21). Zimmermann llama a esta proyección *proyección generalizada* porque si se utiliza una fución de ponderación que asigne uno para todos los píxeles negros y cero a los blancos, se tiene una proyección normal.

Experimentación

En la tabla 4.2 se presentan los resultados de aplicar distintos métodos de corrección del ángulo de *slant* para los corpus ODEC y IDIAP. El método de *slope* utilizado en esta experimentación es el que mejor resultado ha obtenido en el apartado anterior, el método basado en proyecciones horizontales con la función objetivo *máximo*. El método utilizado para normalizar el tamaño del texto es el de *escalado de ascendentes y descendentes*. El mejor resultado para la tarea de ODEC es de 26 WER, mientras que para la tarea de IAMDB es de 26,2, ambos con el método de corrección basado en proyecciones verticales con función de optimización *desviación típica*. Las mejoras relativas obtenidas con respecto a no realizar corrección del *slant* son 27,4 % y 48,8 % respectivamente.

Algoritmo 7 Corrección del Slant: función objetivo Método Zimmermann.

input: $I_{ent}(x, y)$
output: α

for all $\alpha \in [-45, 45]$ **do**
  $I_\alpha \leftarrow Shear(I_{ent}, \alpha)$
end for
$Proj(\alpha, x) \leftarrow 0 \quad \forall \alpha \in [-45, 45] \wedge \forall x$
for all $\alpha \in [-45, 45]$ **do**
  for all x **do**
    $cont \leftarrow 0$
    for all y **do**
      if $(y, x) \in I_\alpha =$ BLACK **then**
        $cont++$
      else
        $Proj(\alpha, x) \leftarrow Proj(\alpha, x) + cont$
        $cont = 0$
      end if
    end for
    $Proj(\alpha, x) \leftarrow Proj(\alpha, x) + cont$
  end for
end for
for all $\alpha \in [-45, 45]$ **do**
  $Z(\alpha) \leftarrow 0$
  for all x **do**
    $Z(\alpha) \leftarrow Proj(\alpha, x)$
  end for
end for
return $\underset{\forall \alpha \in [-45,45]}{\text{argmax}} \; (Z(\alpha))$

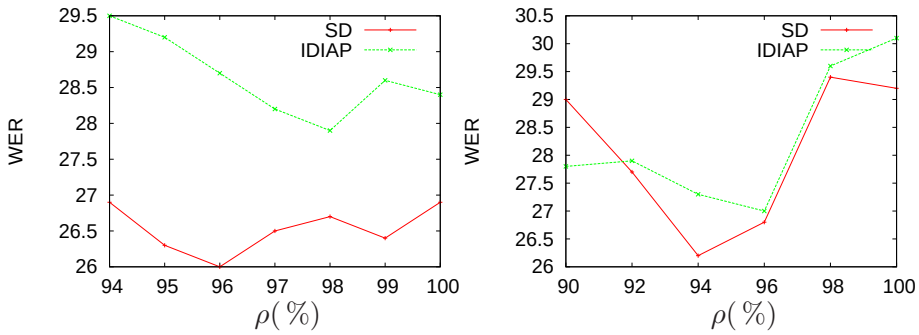


Figura 4.22: Resultados comparativos de test en función del parámetro de suavizado ρ para las funciones objetivo *desviación estándar* e IDIAP. En el panel de la izquierda se muestran los resultados para el corpus ODEC; en el panel de la derecha, se muestran los resultados para el corpus IAMDB.

El método basado en proyecciones verticales con función de optimización *desviación típica* con suavizado mejora los resultados de los demás métodos. El principal inconveniente de este método es la necesidad de ajustar el parámetro ρ. Aun así se observa una significativa mejora al suavizar el método tanto para la función objetivo *desviación típica* como para la *IDIAP* (ver figura 4.22).

Es importante resaltar la mejora relativa obtenida para el corpus IAMDB, debida a la gran diferencia en el ángulo de *slant* entre los diferentes escritores. En cuanto al corpus ODEC, en su mayoría está formado por letras mayúsculas, con lo que el ángulo de *slant* no es tan determinante como en el caso de IAMDB.

método	ODEC	IAMDB
Sin corr. Slant	35.8	51.2
Detección de Bordes	28.2	28.6
IDIAP	28.4	30.1
IDIAP Suavizado	27.9	27.0
Desviación típica	26.9	29.2
Desviación típica Suav	**26.0**	**26.2**
Long. del perfil	27.1	28.1
Zimmermann	28.7	27.6

Tabla 4.2: Tabla comparativa para distintos métodos de estimación del ángulo de *slant* y para los corpus ODEC y IAMDB. Los resultados presentados corresponden a los valores WER obtenidos (ver sección 2.3.3). En el panel superior de la tabla, resultado para el método estructural. En el panel central, resultado para el método de detección de bordes. En el panel inferior, resultados para los métodos basados en proyecciones verticales.

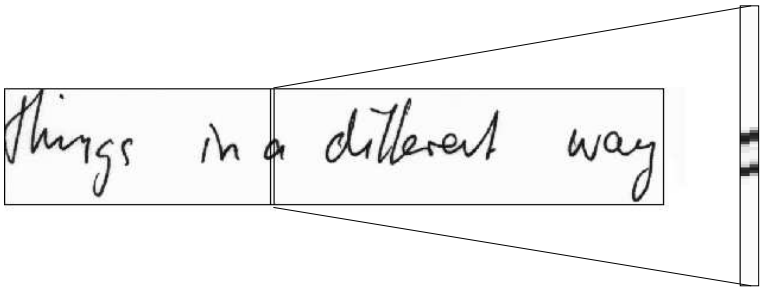


Figura 4.23: Se puede apreciar en la columna extraída la poca información que contiene respecto al texto y gran cantidad de fondo que abarca. Este efecto es debido principalmente a la influencia de los ascendentes y descendentes.

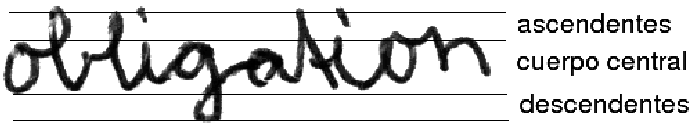


Figura 4.24: Las tres zonas en las que se divide un texto

4.2.3. Normalización del tamaño

La imagen de texto está compuesta por píxeles pertenecientes a dos clases disjuntas, los píxeles del primer plano y los del fondo. La información del texto está contenida en los píxeles del primer plano.

En la fase de extracción de características (que se verá de manera detallada en el capítulo 5), la imagen es segmentada en una serie de columnas. Cada columna se convertirá en un vector que representará la información del área de imagen que cubre la columna. Interesa encuadrar el máximo de área informativa, minimizando la cantidad de píxeles de fondo. El texto se encuadrará en cajas de inclusión mínima, de esta manera, la extracción de características se realizará sobre la zona que contiene la información.

La utilización de HMM para modelar el texto manuscrito (ver capítulo 2) permite un modelado eficiente respecto a las distorsiones horizontales, pero presenta una robustez limitada en cuanto al modelado de la distorsión vertical.

De lo expuesto anteriormente se extrae que la minimización de las zonas poco informativas permite un mejor aprovechamiento de la capacidad expresiva de los modelos morfológicos (HMMs en este trabajo), además de la necesidad de compensar la poca flexibilidad vertical de dichos modelos.

En la escritura se distinguen tres zonas (ver fig. 4.24): el *cuerpo central*, la *zona de ascendentes* que va desde la parte superior del cuerpo central hasta la parte más alta del texto, y la que va desde la parte inferior del cuerpo central a la parte

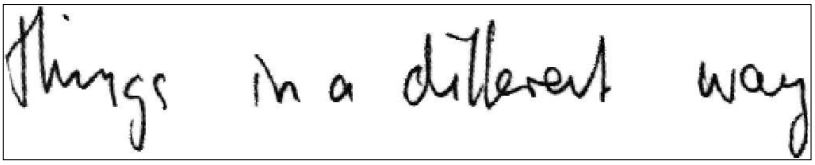


Figura 4.25: En la imagen se aprecia la gran cantidad de espacio en blanco que comprende la caja de inclusión mínima a causa de los ascendentes y descendentes.

más baja del texto, la *zona de descendentes*. Los ascendentes son aquellas partes de ciertos caracteres que sobresalen por la parte superior de la zona del cuerpo central. Caracteres que contiene ascendentes son por ejemplo la *t,l,f*, o las mayúsculas en los nombres o después de un punto. Del mismo modo, los descendentes son aquellas partes de caracteres que sobresalen por debajo del cuerpo central del texto. Los ascendentes y descendentes fuerzan a que se incluyan zonas poco informativas en el encuadre del texto a analizar (ejemplo de la figura 4.25).

La determinación de las distintas zonas del texto no se hace de manera global para cada linea de texto, sino que la linea de texto se divide en segmentos de frase y cada uno de estos segmentos se normaliza de manera individual. Cada segmento de frase está separado de sus vecinos por un gran espacio en blanco. Un segmento de frase no tiene por que ser una palabra, sino que son secuencias de caracteres que están muy cercanos entre sí. De ese modo un segmento de frase puede estar formado por una palabra, una secuencia de ellas o por una parte de palabra. Se asume que si están suficientemente cercanos entre sí es porque se escribieron de manera continua, como una unidad, compartiendo todas las características del estilo de escritura, entre ellas el tamaño de los caracteres. En este trabajo, la estimación de la longitud adecuada del hueco entre segmentos se obtiene a partir de una proyección vertical del texto, de la cual se calcula el tamaño de hueco medio. Este tamaño de hueco medio más una constante empírica determinará los puntos de corte entre tramos.

Una vez umbralizado el segmento de frase, se suaviza con el algoritmo de suavizado por longitud de píxeles consecutivos *Run-Length Smoothing Algorithm*, (RLSA) [WCW82] (ver algoritmo 1). De la imagen obtenida tras aplicar el algoritmo RLSA, se obtiene el contorno superior e inferior del texto suavizado. Básicamente se obtiene el primer y el último píxel negro de cada columna. Los trazos horizontales que se pueden dar en ascendentes y descendentes suelen causar que el algoritmo devuelva un contorno erróneo (ver figura 4.14). Con una sencilla variación del algoritmo se pueden evitar estos problemas. La variación (vista previamente en la sección 4.2.1) consiste en realizar un RLSA tanto horizontal como vertical, para posteriormente localizar por columna el tramo de píxeles negros consecutivos de mayor longitud. Los píxeles superior e inferior de este tramo se tomarán como fronteras para dicha columna.

Para localizar el cuerpo central se ajusta una recta a cada uno de los contornos, previa eliminación de los puntos anómalos causados por ascendentes y descenden-

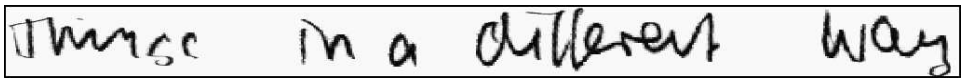


Figura 4.26: Ejemplo de normalización del tamaño de los ascendentes y descendentes, para el texto de la figura 4.25. En la imagen se aprecia la reducción de la zona no informativa, aunque se sigue apreciando una gran desigualdad en el tamaño de los caracteres.

tes. Para los puntos de cada contorno se obtiene la media y la desviación típica. Todos aquellos puntos cuyas y's estén entre el valor de la media más menos la desviación típica serán tomados en cuenta para el ajuste de los bordes superior e inferior de la zona central. Se ajustan a los puntos de cada contorno una recta, por vector propio, por mínimos cuadrados [DH74], o buscando la fila que produce el máximo en la proyección horizontal de cada contorno. Este último método es rápido y barato, y funciona razonablemente bien debido a que al texto se le ha corregido previamente el *slope*.

El límite superior de la zona de ascendentes se calcula buscando en la proyección horizontal aquella fila, empezando por la columna para la cual se tenga el máximo de la proyección, y de manera ascendente localizar aquella fila para la cual se obtiene un valor inferior a un umbral dado. El umbral suele ser 0 que corresponde a columnas con lineas en blanco. El límite inferior de la zona de descendentes se calcula de forma análoga.

Escalado de ascendentes y descendentes

Una primera aproximación para minimizar el área no informativa de la imagen podría consistir en utilizar sólo la zona del cuerpo central. El principal inconveniente es la pérdida de la información que nos permite discriminar por ejemplo entre una *l* y una *e* o entre una *o* y una *p*.

En una primera aproximación, y con el fin de reducir el área poco informativa que rodea al cuerpo central y al mismo tiempo mantener la información de los ascendentes y descendentes, se procederá a escalar la zona de ascendentes y descendentes con respecto a la altura del cuerpo central. La zona central quedará en su tamaño original, asumiendo que un mismo escritor no variará de manera sensible el tamaño de los caracteres. En el caso de que diferentes escritores presenten distinto tamaño de caracteres, no habría de ser ningún problema ya que de la manera que se extraen las características, se produce un escalado implícito. Cada vector vertical de características se obtiene centrando una columna de celdas sobre el texto. Esta columna tendrá el mismo número de celdas independientemente de la altura del texto. Las características se obtienen para cada celda. Para más detalles ver el capítulo 5.

Escalado de las tres zonas del texto

Aunque la normalización del cuerpo central se haga de manera implícita en la fase de extracción de características, la normalización se hace de manera global sobre toda la línea de texto, mientras que la normalización del tamaño del texto se hace por segmentos de frase. En el caso de que un escritor varíe el tamaño de los caracteres en un mismo texto, cosa más habitual de lo que parece razonable, se produce un desajuste en cuanto a la altura de la zona de ascendentes. Al encerrar el texto en una caja de inclusión mínima se reproduce el problema del espacio no informativo que queda englobado dentro. Si bien el espacio no informativo se ha reducido, puede ser reducido más aun simplemente escalando todos los segmentos de frase de manera que tengan el mismo tamaño.

Una vez determinadas las tres zonas para cada segmento de frase, se ha de determinar la altura a la que se van a escalar todos los cuerpos centrales. Hay que decir que la longitud del tramo se escala en la misma proporción que el cuerpo central para mantener la relación de aspecto. Surge la necesidad de tener un criterio para escalar la zona central. Se han probado cuatro criterios para determinar la mejor altura del cuerpo central [Rom06, RPTV06].

- Máximo: La altura elegida para normalizar las zonas centrales de todos los segmentos de frase, es la altura del cuerpo central máxima de todos los segmentos. Este método presenta dificultades en los casos en que el texto presente mayúsculas o tramos cortos con muchos ascendentes y/o descendentes cosa que suele producir una incorrecta determinación de las tres zonas del texto. Estos problemas hace que algunos segmentos de frase se amplíen demasiado deformándose.

- Media: La altura seleccionada será la media de la altura de los cuerpos centrales de todos los segmentos de frase.

- Media Ponderada: Con la intención de deformar lo menos posible la imagen, los segmentos de mayor longitud tendrán mayor peso que los segmentos cortos. Para ello, la altura de cada segmento, A_i se ponderará con su longitud, p_i. La altura a la que se escalarán todos los segmentos se calcula de la siguiente manera:

$$\overline{A} = \sum_{i=1}^{N} p_i \cdot A_i \tag{4.37}$$

$$p_i = \frac{l_i}{\sum_{j=1}^{N} l_j}$$

  donde N es el número de segmentos del texto y l_i es la longitud del segmento i.

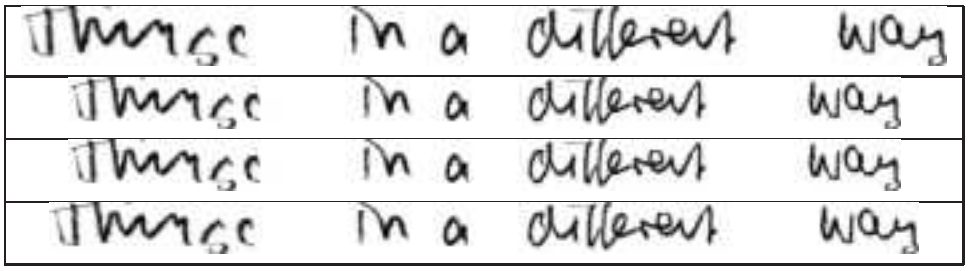


Figura 4.27: Ejemplos de normalización del tamaño de la figura 4.25 con diferentes criterios para la elección del tamaño del cuerpo central. De arriba a abajo: criterio *máximo*, criterio *media*, criterio *media ponderada* y criterio *moda con contexto*.

- Moda con contexto: Con la idea de modificar el menor número de segmentos de frase posibles, se toma como altura de normalización aquella que sea más frecuente entre los segmentos de frase, esto es, la moda de alturas. El histograma de alturas de segmentos de frase se puede suavizar para evitar saltos abruptos. El suavizado planteado aquí fuerza, además de que se modifiquen el menor número de segmentos, que los que se modifiquen se modifiquen lo menos posible. Para este suavizado se tendrá en cuenta el contexto de cada punto del histograma. Cada entrada del histograma suavizado se calculará como la suma de las frecuencias incluidas en una ventana centrada en cada punto del histograma original (ecuación 4.38).

$$\overline{H(i)} = \sum_{j=i-K}^{i+K} H(j) \tag{4.38}$$

  La altura de normalización se tomará como el valor más frecuente en el histograma normalizado $\overline{A} = \underset{0 \leq i \leq |\overline{H}|}{\text{argmax}}\, \overline{H}(i)$.

Experimentación

En la tabla 4.3 se presentan los resultados experimentales obtenidos dependiendo de los diversos criterios para normalizar el tamaño de los caracteres. Los resultados se presentan para los corpus ODEC e IAMDB. Los métodos de normalización del ángulo de *slope* y de *slant* utilizados son los que mejor resultando han obtenido en las secciones previas: función objetivo *maximo* para método basado en proyecciones horizontales para el *slope* y función objetivo *desviación típica* suavizada con un factor $\rho = 98\,\%$ para el *slant* .

método	ODEC	IAMDB
Baseline	31.2	58.6
Escalado de asc. y des.	26.0	26.2
Máximo	27.1	29.6
Media	26.0	28.4
Media Ponderada	**25.8**	**25.6**
Moda con contexto	26.7	25.9

Tabla 4.3: Tabla comparativa para los distintos métodos de normalización del tamaño y para los corpus ODEC e IAMDB. Los valores representados corresponden con los valores WER obtenidos (ver sección 2.3.3). En el panel superior de la tabla se muestra el resultado obtenido para el caso que no se normalize el tamaño; en el panel central, resultado para el caso de que sólo se escalen los ascendentes y descendentes. En el panel inferior, resultados para los distintos criterios de selección del tamaño del cuerpo central.

Para el corpus ODEC se aprecia una gran mejora relativa (entorno al 17 %) al normalizar el tamaño con el método que mejor resultado obtiene, respecto de no normalizar. El mejor resultado se obtiene al utilizar el criterio de selección del tamaño del cuerpo central *Media Ponderada*. Hay que notar que para este corpus el escalado de ascendentes y descendentes reporta un resultado próximo al mejor.

El criterio *Media Ponderada* produce el mejor resultado tambien para el corpus IAMDB. En este caso la mejora relativa con respecto de no normalizar es de 56.3 %, esta gran mejora es debida a la gran diversidad de tamaño de caracteres que hat en este corpus, sobre todo en lo que atiende a ascendentes y descendentes.

4.3. Resumen

El estilo de escritura no aporta nada a los sistemas de RATM, ya que lo escrito no tienen nada que ver con el estilo en que fue escrito. El texto manuscrito presenta una gran diversidad de estilos de escritura. La normalización pretende dotar de cierta invariabilidad a los sistemas RATM frente a los distintos estilos de escritura.

El preproceso consiste en una serie de transformaciones sobre la señal original con la intención de obtener la máxima homogeneidad posible dentro de cada clase. El objetivo del preproceso es ayudar al módulo de extracción de características para que produzca valores para las características lo más parecidos entre sí para patrones que pertenezcan a una misma clase, y lo más distintos posibles para patrones de otras clases. O dicho de otro modo, incrementar la robustez del sistema RATM frente a la entrada de texto manuscrito.

Los métodos de normalización se han estructurado en dos grandes grupos, los métodos que normalizan el texto a nivel de página, y los que lo hacen centrándose en el texto contenido en la página. En el primer grupo se han estudiado técnicas

globales y adaptativas de umbralización de la imagen, métodos de reducción del ruido, y de corrección del desencuadre o *skew*. En el segundo grupo de métodos se han estudiado técnicas de corrección del ángulo de *slant*, y de normalización del tamaño del texto.

En este capítulo se han estudiado y modificado importantes técnicas de normalización de la señal de entrada, al mismo tiempo que se han propuesto nuevos métodos. También se han formalizado las soluciones basadas en proyecciones, como problemas de optimización donde cada método propuesto en la literatura puede verse como una función objetivo. Se han probado y modificado algunas de las funciones más prometedoras de la literatura, y se han propuesto nuevas funciones.

CAPÍTULO 5

EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características establece un nuevo espacio de representación de la señal de entrada. Este nuevo espacio debe permitir una representación compacta de la señal, y debe facilitar la discriminación entre las distintas clases de patrones, en el sentido de que se minimice la variabilidad intra-clase, al mismo tiempo que maximiza la variabilidad iter-clase. Esto implica que los valores que tomen las características para las muestras de una clase presenten la menor variabilidad posible, al mismo tiempo que son distantes para el resto de muestras de las otras clases. Además, las características deben ser suficientemente invariables para que estén presentes en cualquier estilo de escritura

La elección del conjunto características es una decisión crítica que depende de la tarea y del clasificador que se use. El mejor conjunto de características para una tarea dada será aquel, entre aquellos adecuados para el tipo de reconocedor usado, que proporcione la máxima precisión con el mínimo número de características. Además, es deseable que la extracción de características se obtenga de manera sencilla y a un bajo coste en recursos. Por lo tanto, un buen diseño debe maximizar la precisión del sistema y minimizar su tiempo de respuesta .

En la literatura se pueden encontrar métodos formales para seleccionar características, aunque la intuición sigue siendo la técnica más popular y generalmente efectiva [Ste85]. La elección del mejor conjunto de características se suele hacer a partir de una evaluación experimental. Si se utiliza un clasificador estadístico y se dispone de muchas muestras, se puede utilizar *análisis discriminante* [CY97] para seleccionar aquellas características que posean un mayor poder discriminativo y reducir la dimensionalidad del espacio.

En este capítulo se presenta una breve revisión de los diferentes tipos de características utilizadas en el campo del reconocimiento automático de texto, para luego centrarse en las que se utilizarán a lo largo de este trabajo.

5.1. Taxonomía de las características

El caso más básico de conjunto de características consiste en tomar los própios píxeles de la imagen entera. Los principales problemas de esta elección son su alta sensibilidad al ruido y a cualquier tipo de distorsión en los caracteres, además de la gran dimensionalidad.

En la literatura se suelen encontrar cientos de características usadas en el campo de RATM [Ari98, ØDT96]. Una posible categorización de las distintas características se presenta a continuación:

a) Extraídas a partir de la distribución estadística de los puntos. Este tipo de representaciones no permiten la reconstrucción, ni parcial ni total, de la imagen original. Este tipo de características es muy usado debido a la sencillez y baja complejidad computacional de los algoritmos usados para obtenerlas. Otra gran ventaja es que suelen tener una dimensionalidad baja, lo que permite a los sistemas tener buenos tiempos de respuesta. Entre este tipo de características cabe destacar:

   - *Zoning*: la imagen de nivel de gris se divide en $n \times m$ celdas, que pueden estar solapadas o no. Para cada celda se calcula el nivel de gris, la dirección del gradiente, la dirección del contorno del caracter, o cualquier otro tipo de distribución de los píxeles [MM94].
   - Proyecciones: las imágenes se representan mediante proyecciones de los niveles de gris en los ejes de coordenadas. Este tipo de características crea representaciones unidimensionales a partir de las dos dimensiones de la imagen [TT99, WTS88].
   - Cruces y distancias: las características consisten en el número de veces que se cruza el contorno del texto con una serie de lineas predefinidas en determinadas direcciones, y/o el conjunto de las distancias de los puntos del contorno con respecto a ciertas referencias, como por ejemplo los bordes de las cajas de inclusión mínima [AV00, MG96].

b) Extraídas a partir de transformaciones globales y expansión de series. Una manera compacta de representar una señal es mediante combinaciones lineales de series de funciones. Los coeficientes de la combinación lineal es una forma compacta de codificar la señal. Una ventaja destacable de este tipo de representación es que permite reconstruir la imagen a partir de las características. A continuación se citan algunas de las transformaciones y series de expansión utilizadas en el campo del reconocimiento de texto.

   - Transformada Fourier: cualquier función periódica puede ser expresada como una suma infinita de senos y cosenos de distintas frecuencias.

     Como las imágenes son señales bidimensionales discretas se necesita una transformada discreta de Fourier de dos dimensiones, lo cual implica que se obtendrán dos dimensiones de la frecuencia: horizontal u

y vertical v. Suponiendo una imagen de dimensiones $N \times M$, la transformada discreta bidimensional sería [Ell87]:

$$FP(u,v) = \frac{1}{\sqrt{NM}} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} I(x,y) e^{-i2\pi(\frac{ux}{N}+\frac{vy}{M})} \qquad (5.1)$$

Algunos ejemplos de utilización de la transformada de Fourier se pueden ver en [WCL94, ZSW99].

- Filtros Gabor 2D: Los filtros de Gabor están relacionados con la forma en que la corteza visual procesa las imágenes. El filtro de Gabor es un filtro lineal cuya respuesta de impulso es una función armónica multiplicada por una función gaussiana [Gab46]. Estos filtros son filtros pasabanda selectivos a la orientación y a la frecuencia . Los filtros constituyen un banco donde cada uno de ellos representa versiones dilatadas, trasladadas y rotadas respecto a las de los demás. La forma general de un filtro de Gabor está definida por la siguiente formula:

$$h(x,y,\Theta_k,f,\sigma_x\sigma_y) = e^{-\frac{1}{2}(\frac{X^2_{\Theta_k}}{a^2_x}+\frac{Y^2_{\Theta_k}}{\sigma^2_y})} e^{(i2\pi f X_{\Theta_k})} \quad k = 1,...,m \qquad (5.2)$$

donde $X_{\Theta_k} = x.cos(\Theta_k) + y.sen(\Theta_k)$ y $Y_{\Theta_k} = -x.sen(\Theta_k) + y.cos(\Theta_k)$, f es la frecuencia de la onda sinusoidal plana, m es la cantidad de orientaciones, Θ_k es la k-ésima orientación del filtro de Gabor y donde σ_x y σ_y son las desviaciones estándar de la envolvente Gaussiana a lo largo de los ejes x e y.

- Wavelets discretos: esta es una aproximación reciente. Fue introducida a principios de los noventa por Daubechies [Dau90]. Su principal ventaja es que permite un análisis a diferentes escalas o resoluciones. La transformada discreta wavelet (WDT) de una imagen divide ésta en dos tipos de imágenes: la tendencia y las fluctuaciones. La tendencia suele ser la imagen original a menor resolución, mientras que las fluctuaciones dan cuenta de los cambios locales. No existe una transformada wavelet única, sino que existe una colección de familias wavelet. Algunos ejemplos de utilización de coeficientes wavelets como características en sistemas de RATM pueden encontrarse en [LK95, SWN98].

- Expansiones de Karhunen-Loéve: estas expansiones están basadas en un análisis de vectores própios que permite reducir la dimensión del conjunto de características creando nuevas a partir de combinaciones lineales de las originales. Esta transformación es óptima en el sentido de la compresión de datos, esto es, la misma información contenida en la imagen se representa con un pequeño conjunto de características. El inconveniente de esta representación es la complejidad computacional de los algoritmos necesarios.

A pesar del alto coste computacional, en la actualidad se puede encontrar un sistema comercial de OCR cuya extracción de características está basada en expansiones de Karhunen-Loéve [F'u99].

- Momentos geométricos: fueron introducidos por Hu [Hu62] en 1962. Los momentos son considerados como series de expansión puesto que la imagen original puede ser reconstruida completamente a partir de los coeficientes de los momentos. Los momentos son propiedades numéricas obtenidas de una imagen. Los momentos geométricos pueden ser *simples*, *centrales* o *centrales normalizados*.

  Para una imagen de niveles de gris I(x,y) compuesta por M píxeles, los *momentos geométricos* simples de orden $(p+q)$ se definen como:

$$m_{pq} = \sum_{i=1}^{M} I(x_i, y_i)(x_i)^p (y_i)^q \tag{5.3}$$

  Los *momentos centrales* de una imagen son sus *momentos geométricos* tomando como origen de coordenadas su centro de masa $(\overline{x}, \overline{y})$:

$$\mu_{pq} = \sum_{i=1}^{M} I(x_i, y_i)(x_i - \overline{x})^p (y_i - \overline{y})^q \tag{5.4}$$

  donde:

  - $\overline{y} = \frac{m_{01}}{m_{00}}$ coordenada media y del nivel de gris.
  - $\overline{x} = \frac{m_{10}}{m_{00}}$ coordenada media x del nivel de gris.

  Alguna de las propiedades destacables de los *momentos centrales* son:

  - μ_{00} es la masa de gris de la imagen.
  - $\mu 01 = \mu 10 = 0$ puesto que el centro de masas está en el origen.
  - Invariantes a la translación.
  - Los momentos centrales de orden 2 ($p + q = 2$) definen un área elíptica que se ajusta a la masa de gris de la imagen. A partir de estos momentos, se pueden definir la *orientación* θ y la *excentricidad* global ϵ de la masa de gris.

$$\theta = atan\left(\frac{\mu_{02} - \mu_{20} - 2\mu_{11} + \lambda}{\mu_{02} - \mu_{20} - 2\mu_{11} - \lambda}\right) \quad \epsilon = \sqrt{\frac{\mu_{02} + \mu_{20} + \lambda}{\mu_{02} + \mu_{20} - \lambda}} \tag{5.5}$$

    donde

$$\lambda = \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \tag{5.6}$$

    Los valores de θ y de ϵ son invariantes a la translación y a la escala. Además el valor ϵ es invariante a la rotación.

Los *momentos centrales normalizados* de una imagen se definen a partir de sus momentos centrales normalizandolos con *momentos* de orden 0:

$$\eta = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \qquad \gamma = \frac{p+q+2}{w} \tag{5.7}$$

c) Características topológicas y geométricas. Diversas propiedades, locales o globales, pueden ser representadas a partir de características topológicas y geométricas. En este tipo de características se puede empotrar conocimiento *a priori* sobre la estructura de los objetos a representar. Además, este tipo de representación presenta gran tolerancia a la distorsión y a los distintos estilos de escritura. En la literatura se pueden encontrar un sinfín de representaciones que pueden ser agrupadas en cuatro grandes grupos:

- Extracción de estructuras topológicas: se intentan localizar estructuras, en el caracter o palabra, a partir de un conjunto de estructuras previamente definidas. El número y/o la posición de estas estructuras conforma una representación descriptiva. El conjunto de primitivas suele estar formado, entre otros, por lineas, arcos, curvas complejas, splines, puntos extremos, cruces por el eje x, lazos, número de máximos y mínimos, aperturas en las cuatro direcciones principales.
- Medidas y aproximación de propiedades geométricas: los caracteres y/o las palabras pueden ser representadas con medidas sobre propiedades geométricas. Algunas medidas muy usadas son por ejemplo la longitud de la palabra, la masa de texto en la parte superior e inferior, la curvatura en cada punto, o en el caso de clasificación de caracteres, el cociente entre el alto y el ancho de la caja de inclusión mínima.
- Codificación de los contornos con códigos de Freeman (o *Chain-Codes*): este es un esquema muy conocido y utilizado [MKG99]. El contorno, o del esqueleto de un caracter, se divide en segmentos del mismo tamaño, cada segmento se codifica por su dirección (usualmente ocho direcciones). En la codificación de cada caracter o palabra se usan tres tipos de información: el punto inicial, la longitud del segmento y un vector con las direcciones de cada segmento.
- Grafos: cada caracter o palabra se representa en forma de grafo de primitivas [LOHG97, LRS91]. Esta representación requiere que las palabras y/o los caracteres se dividan en primitivas, como trazos, cruces, lazos, etc. La relación entre las primitivas que conforman un caracter o una palabra se representa mediante un grafo. Los árboles, como caso particular de grafos, son muy utilizados en aquellas lenguas cuyas palabras están compuestas por primitivas que pueden ordenarse de manera jerárquica [JLN03], como la mayoría de lenguas asiáticas.

Figura 5.1: A la imagen se le aplica una ventana deslizante que irá barriendo la imagen de izquierda a derecha. Para cada superficie de la imagen cubierta por la ventana deslizante en cada posición horizontal se extraerán una serie de características.

5.2. Características usadas en esta tesis

Como se ha visto con anterioridad, el conjunto de características es dependiente del clasificador que se utilice. En esta tesis se ha utilizado un reconocedor basado en modelos ocultos de Markov, qué modelan secuencias de vectores. Se necesita pues convertir una imagen bidimensional en una secuencia de características. La naturaleza del texto incluye un orden temporal, el texto se escribe de izquierda a derecha, pero en el caso de texto *off-line* la información temporal se ha perdido. Se pretende aproximar la relación temporal del texto mediante una ventana deslizante, que irá cubriendo la imagen que contiene el texto, de izquierda a derecha (ver ejemplo de la imagen 5.1). Siguiendo el trabajo de Makhoul [MSLB98], para cada posición sucesiva de la ventana se obtendrán una serie de características sustentadas en mediciones sobre la distribución de los puntos. La imagen quedará representada por la secuencia pseudotemporal de vectores de características.

En el caso que atañe a esta tesis, la superficie de la imagen cubierta por ventana deslizante se divide horizontalmente en N celdas iguales. De la superficie de la imagen cubierta por cada una de estas celdas se obtienen tres características: *nivel de gris medio normalizado* y las *derivadas horizontales* y *verticales*. El nivel de gris explica la densidad del trazo en la zona analizada, mientras que las derivadas verticales y horizontales explican como varía esta densidad con respecto a los ejes de coordenadas.

Para cada posición de la ventana deslizante se obtendrá un vector de características. La ventana deslizante tiene dos parámetros: el ancho de la ventana (el alto viene determinado por la altura de la imagen), y el factor de solapamiento. El ancho de la ventana se toma como una porción de la altura (como por ejemplo 1/16, 1/20, 1/24 o 1/28). A este valor se le denomina *resolución de la extracción de características*. Como el número de celdas N se elige para que éstas sean cuadradas (aunque no tendrían porque serlo, es lo más usual), N se toma como el denominador de la *resolución de la extracción de características*. Siguiendo el trabajo de Toselli [Tos04] la resolución elegida es 1/20 y el factor de solapamiento es 0.

El cálculo de las características de cada celda no se restringe solamente a dicha

celda, sino que se extiende a su contexto. Para ello, a cada celda de la ventana deslizante se le superpone una ventana de análisis de tamaño mayor. La ventana de análisis se ha elegido, siguiendo también el trabajo de Toselli, de un tamaño cinco veces el tamaño de la ceda.

5.2.1. Nivel de gris normalizado

Para cada celda de la ventana deslizante se calcula el nivel de gris normalizado. El cálculo del nivel de gris se extiende al contexto de la celda mediante la superposición de una ventana de análisis centrada sobre la celda. Supóngase una ventana de análisis $A(x, y)$ de $m \times m$ píxeles:

$$A(x,y) = \begin{pmatrix} (x_1,y_1) & \cdots & (x_m,y_1) \\ \vdots & & \vdots \\ (x_1,y_m) & \cdots & (x_m,y_m) \end{pmatrix} \tag{5.8}$$

Para enfatizar el rol de los píxeles centrales y disminuir la influencia de los más lejanos, se pondera cada píxel de la ventana de análisis mediante una gausiana bidimensional:

$$\hat{A}(x,y) = A(x,y) \exp\left[-\frac{1}{2}\left(\frac{(x-\frac{n}{2})^2}{(n/4)^2} + \frac{(y-\frac{m}{2})^2}{(m/4)^2}\right)\right] \tag{5.9}$$

El valor de gris normalizado para la celda sobre la que está centrada la ventana de análisis se calcula del siguiente modo:

$$g = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} \hat{A}(x_i, y_j)}{m\,m} \tag{5.10}$$

En la imagen 5.2 se puede ver de manera gráfica un ejemplo del proceso de obtención del nivel de gris para una celda dada.

5.2.2. Derivada vertical y horizontal

Las derivadas dan cuenta de como varía la densidad de gris con respecto a cada eje de coordenadas. Aunque sólo con los niveles de gris normalizados se tiene información completa de toda la imagen, la inclusión de las derivadas compensa la asunción inherente a los modelos de Markov de independencia condicional entre las distintas muestras (vectores de características) [MSLB98].

La derivada horizontal se calcula como la pendiente de la linea que mejor se ajusta a la función de nivel de gris medio por columna (ecuación 5.11). El criterio de ajuste de la recta es la minimización del error cuadrático (ecuación 5.12). Para

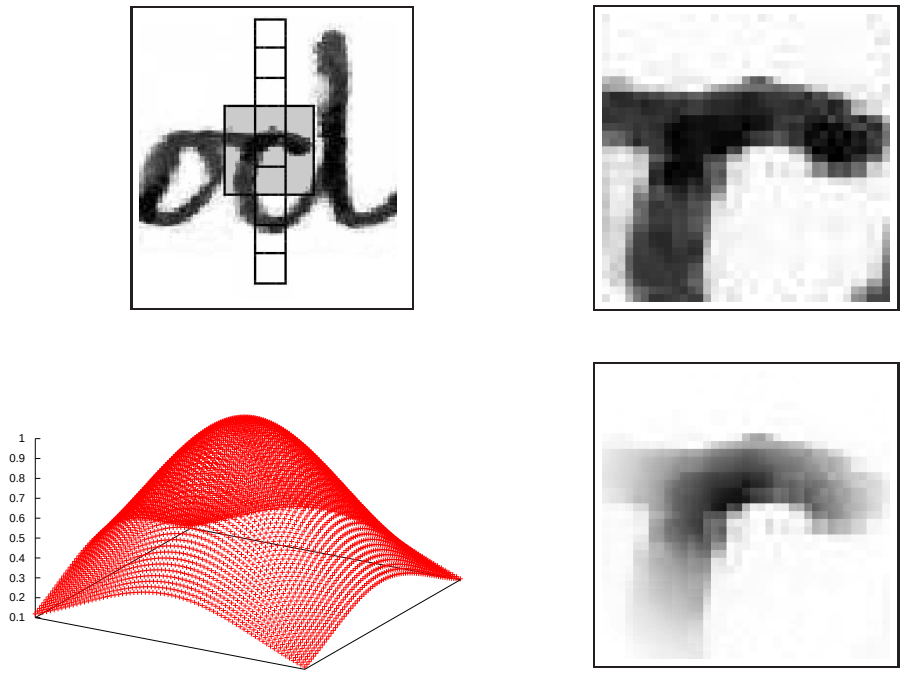


Figura 5.2: Ejemplo de extracción del *nivel de gris normalizado*. A la ventana de análisis (arriba izquierda y derecha) se le aplica un filtro gausiano bidimensional que realza la importancia de los píxeles centrales y atenúa la de los más lejanos. El nivel de gris para la celda central se obtendrá como el promedio de gris normalizado de la ventana de análisis suavizada (abajo derecha).

dar mayor peso a las columnas centrales se aplica un filtro gausiano unidimensional. La derivada vertical se calcula de manera similar. El nivel de gris normalizado para cada columna $g(x_i)$ se calcula del siguiente modo:

$$g(x_i) = \frac{\sum_{j=1}^{m} \hat{A}(x_i, y_j)}{m} \tag{5.11}$$

A la función $g(x_i)$ se le ajusta una recta $ax + b$ mediante regresión lineal, cuya distancia a cada uno de los puntos $(x_i, g(x_i))$ minimiza la función objetivo:

$$J = \sum_{i=1}^{m} w_i (g(x_i) - (ax_i + b))^2 \tag{5.12}$$

Para enfatizar la aportación de los puntos centrales de la función $g(x_i)$, el error cuadrático se pondera mediante un filtro gausiano unidimensional:

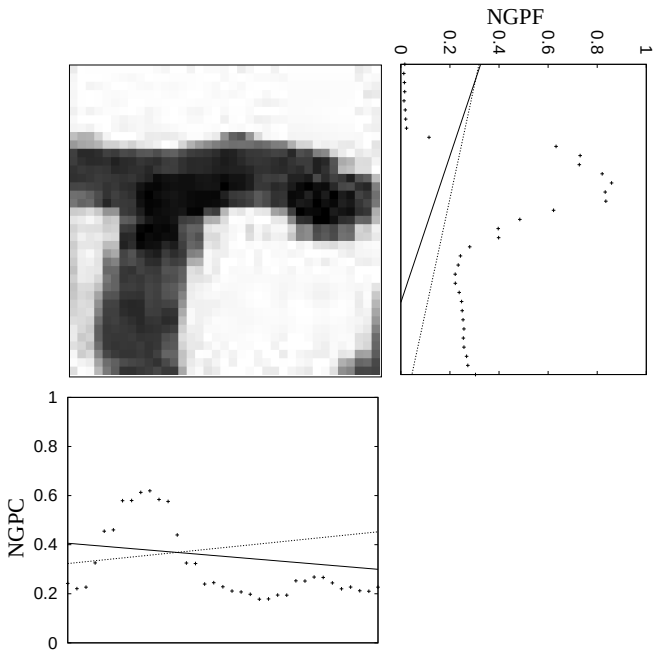


Figura 5.3: Ejemplo de cálculo de la derivada horizontal (panel inferior) y vertical (panel derecho). Las derivadas se calculan como la pendiente de la recta que mejor se ajusta a los puntos de *nivel de gris promedio por fila* (NGPF), en el caso de la derivada vertical, y a los puntos de *nivel de gris promedio por columna* (NGPC), en el caso de la derivada horizontal. Ambas rectas se ajustan por mínimos cuadrados (líneas punteadas). Para priorizar la aportación de los píxeles centrales, el ajuste se hace ponderado por un filtro gausiano (líneas de trazo grueso).

$$w_i = \exp\left(-\frac{1}{2}\frac{(x_i - \frac{m}{2})^2}{(\frac{m}{4})^2}\right) \tag{5.13}$$

derivando la función objetivo 5.12 respecto a a (pendiente) y a b (valor de la ordenada al origen de la recta) e igualando cada expresión a 0 se obtiene el valor de a (expresión 5.14), o lo que es lo mismo la dirección dominante del nivel de gris dentro de la ventana de análisis.

$$a = \frac{\sum_{i=1}^{m} w_i\, g(x_i) \sum_{i=1}^{m} w_i\, x_i \;-\; \sum_{i=1}^{m} w_i \sum_{i=1}^{m} w_i\, g(x_i)\, x_i}{\left(\sum_{i=1}^{m} w_i\, x_i\right)^2 \;-\; \sum_{i=1}^{m} w_i \sum_{i=1}^{m} w_i\, x_i^2} \tag{5.14}$$

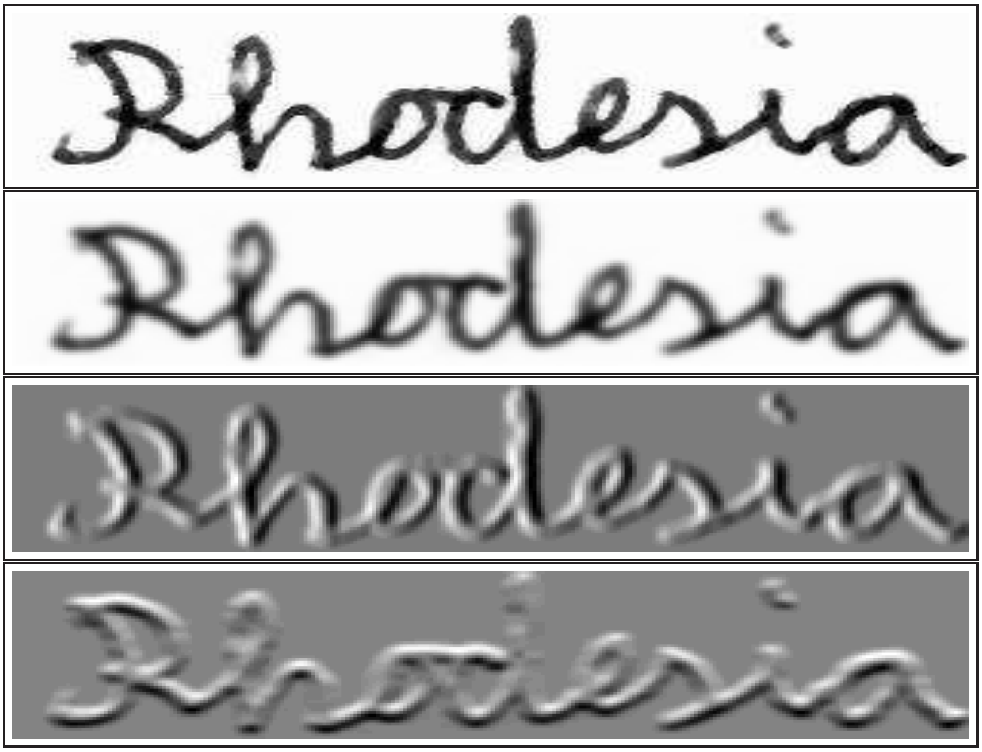


Figura 5.4: Ejemplo gráfico del proceso de extracción de características. De arriba a abajo: imagen en niveles de gris preprocesada, representación gráfica de la característica *niveles de gris normalizados*, representación gráfica de las *derivadas horizontales*, y representación gráfica de las *derivadas verticales*.

CAPÍTULO 6

ADAPTACIÓN AL ESCRITOR

El conjunto de muestras con el que se estiman los parámetros de los modelos morfológicos suele estar compuesto de textos producidos por diferentes escritores (modelos *independientes del escritor*, IE). De esta manera, el sistema responde razonablemente bien, incluso para escritores para los que no se disponía de muestras en la fase de entrenamiento. Estos modelos no son óptimos para ningún escritor, pero son suficientemente buenos para todos ellos. Ajustar los parámetros de los modelos IE para un escritor concreto, produce nuevos modelos cercanos a los óptimos para él que serían los modelos entrenados sólo con muestras de dicho escritor (modelos *dependientes del escritor*, DE). Los datos necesarios para adaptar modelos IE a un escritor concreto, son muchos menos que si el entrenamiento fuera DE.

Muchos de los sistemas comerciales de reconocimiento de texto manuscrito suelen ser utilizados por una única persona. Lo ideal sería disponer de suficientes muestras de este único escritor para construir un sistema ajustado a su forma de escribir. Ahora bien, obligar a un escritor a producir suficientes muestras para entrenar el sistema resulta impensable debido al esfuerzo que tendría que hacer antes de que el reconocedor fuese operativo. Una solución a este problema consiste en tener sistemas generales IE, que funcionen razonablemente bien para cualquier escritor, y adaptarlos a un escritor concreto, de manera que con poco esfuerzo por su parte, se mejore la productividad original del sistema para dicho escritor.

Si la adaptación se realiza a partir de un conjunto de muestras previamente adquiridas, se la denomina *adaptación estática*, mientras que si la adaptación se realiza de manera incremental durante el uso del sistema, se la denomina *adaptación dinámica* o *incremental*. Por otro lado, si la adaptación se hace a partir de muestras etiquetadas se la denomina *adaptación supervisada*, y en el caso de que no estén etiquetadas, *adaptación no supervisada*.

Estas técnicas han sido utilizadas con éxito en el campo del reconocimiento automático de voz [LW95a, LW95b, GW96, Gal96, Gal00, DRN95, DCB+99]. Hay dos tipos de aproximaciones [Chr96], la primera de ellas busca transformar el espectro de la señal de las muestras del nuevo usuario para que se ajuste lo máximo posible al de la señal con la que se entrenó el sistema. La otra aproximación pretende ajustar los modelos originales HMM para que se adapten mejor al los nuevos

datos de adaptación. En este capítulo se estudia la adaptación estática y supervisada de HMMs continuos, entrenados de manera independiente del escritor, con la intención de aumentar la precisión del sistema para un escritor concreto.

El capítulo se estructura de la siguiente manera: en la sección 6.1 se introducen las técnicas de adaptación. En la sección 6.2 se expone la técnica de adaptación denominada MLLR (*Maximum Likelihood Linear Regression*). En la sección 6.3 se expone el estudio experimental. En la última sección se presenta un resumen del capítulo.

6.1. Técnicas de adaptación al escritor

Para un modelo HMM previamente entrenado, con parámetros $\Theta = \{\theta_1, \cdots, \theta_N\}$ y para un conjunto de observaciones O_d, se pretende transformar los parámetros del modelo Θ a través de una función $F_W(\cdot)$, que permita obtener un modelo adaptado Θ_W, donde la representación del escritor que ha producido O_d sea mayor (ver figura 6.1). Sea W el conjunto de parámetros de la transformación, se quiere encontrar un conjunto $\hat{W}$, de tal manera que:

$$\hat{W} = \underset{W}{\text{argmax}}\ P(O_d|\Theta, W) \tag{6.1}$$

donde $P(O_d|\Theta, W)$ representa la probabilidad de que los datos de adaptación se obtengan a partir los modelos Θ, transformados por una función $F_W(\cdot)$. Así pues $P(O_d|\Theta, W) = P(O_d|F_W(\Theta))$.
Aplicando la regla de Bayes:

$$\hat{W} = \underset{W}{\text{argmax}}\ \frac{P(O_d|\Theta, W)\ P(\Theta, W)}{P(O_d)} \tag{6.2}$$

como la distribución de probabilidad de la muestra, $P(O_d)$ no influye en la maximización, se puede obviar, quedando la ecuación 6.2 como:

$$\hat{W} = \underset{W}{\text{argmax}}\ P(O_d|\Theta, W)\ P(\Theta, W) \tag{6.3}$$

Hay dos técnicas clásicas para realizar esta adaptación, dependiendo de la información sobre la distribución de los parámetros. Si $P(\Theta)$ es no informativa, lo que implica que no se tiene ninguna información acerca de la distribución de los parámetros Θ, entonces los parámetros adaptados Θ_W se estiman mediante el algoritmo de máxima verosimilitud (ML, del inglés *Maximum Likelihood*). En este caso, la distribución de probabilidad, $P(\Theta, W)$ se aproxima por una constante.

Por contra, si se tiene información *a priori* de la distribución $P(\Theta)$, los parámetros adaptados óptimos se pueden obtener con la técnica conocida como *Maximum A Posteriori* o MAP [LG93, GL94]. A la adaptación MAP se la conoce también como Adaptación Bayesiana.

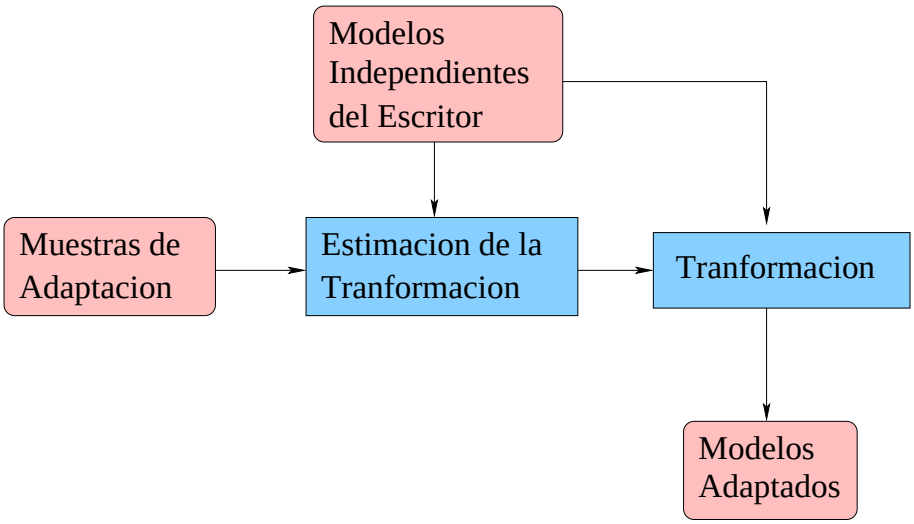


Figura 6.1: Esquema general del proceso de adaptación.

Los modelos morfológicos usados en este trabajo son los muy conocidos HMMs continuos, donde la probabilidad de que una observación se produzca, estando en un estado determinado, está modelada por una mixtura de gausianas (aunque se podrían usar otras distribuciones). Los parámetros de una mixtura de gausianas son las medias, las varianzas y los pesos de cada gausiana dentro de la mixtura, $\Theta = \{\mu_g, \Sigma_g, w_g\}$. Siguiendo la aproximación propuesta por [LW95a, LW95b] se adaptarán solamente los vectores de medias. Así, los parámetros a adaptar serán $\Theta = \{\mu_1, \cdots, \mu_G\}$, siendo G el número de gausianas del modelo. Al no modificar las varianzas, la forma de las distribuciones originales se mantiene inalterada (ver el ejemplo de la figura 6.2).

6.2. *Maximum Likelihood Linear Regression*: MLLR

Esta técnica fue introducida por Leggetter en [LW95a]. Su funcionamiento se basa en calcular una serie de transformaciones lineales, que aplicadas a las gausianas de los modelos HMM, reduzcan la falta de ajuste de estos, con respecto a un conjunto de muestras de adaptación. Este conjunto de transformaciones se compone de desplazamientos de las medias y modificaciones de las varianzas, de tal manera que se incremente la probabilidad de que esas gausianas generen las muestras de adaptación.

La técnica MLLR fue ideada para trabajar con pocas muestras de adaptación. En el caso más básico, donde se tengan muy pocas muestras de adaptación, se puede realizar una adatación global, donde se aplica la misma transformación para todas las gausianas de los modelos. Lo ideal sería disponer de muestras representativas del escritor para cada modelo de caracter. Si no es el caso, se pueden agrupar

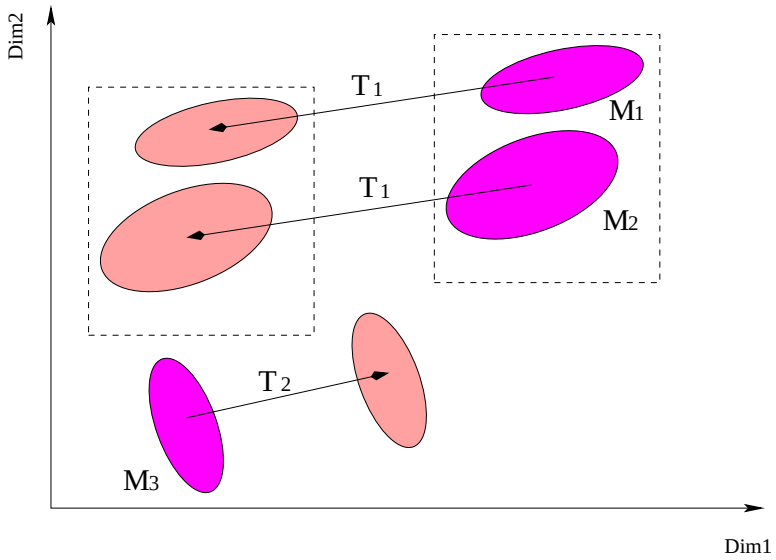


Figura 6.2: La transformación de los vectores de medias tiene el efecto de desplazar las mixturas dentro del espacio de características, pero sin cambiar sus formas. En el ejemplo se muestra el desplazamiento para dos clases de regresión: la formada por las mixturas M_1 y M_2, y la formada por la mixtura M_3, en un espacio de características de dos dimensiones.

las gausianas en clases de regresión, donde se dispondría de suficientes muestras por clase para estimar una transformación para cada clase.

6.2.1. Árboles binarios de regresión

Las clases de regresión se determinan utilizando un árbol de regresión binario. Estos árboles se construyen de forma que agrupen aquellas componentes, en nuestro caso vectores de medias, que estén cercanas en el espacio de las muestras, de modo que las componentes que sean similares se modificarán de la misma manera. El número de clases de regresión dependerá del conocimiento *a priori* que se tenga de la tarea, del número de gausianas de los modelos, y de la cantidad de muestras de adaptación. En el límite, cada clase contendrá una gausiana (su vector de medias). Lo ideal es tener cuantas más clases mejor, pero cuando el número de clases aumenta, la cantidad "necesaria"de muestras necesarias para adaptar el modelo también crece.

El árbol de clases de regresión se suele crear siguiendo uno de los siguientes esquemas:

- Aproximación sencilla:
  - Si se tienen pocos datos construir una única clase de regresión con todas las gausianas.

- Si se tienen suficientes datos construir una clase por gausiana.

- Basado en conocimiento: el conocimiento de la tarea permite decidir como se va a particionar el conjunto de gausianas. Por ejemplo: las correspondientes a mayúsculas, minúsculas, redondeadas, verticales, con ascendentes, etc.

- Dirigido por los datos: los componentes se agrupan de acuerdo a su proximidad en el espacio de las características, de tal manera que componentes similares se modifiquen de manera similar. Una forma usual de dividir las muestras en subconjuntos siguiendo un criterio de proximidad en el espacio de características es la expresada en el siguiente esquema:

  1. Empezar con todos los vectores en la misma clase de regresión.
  2. Para cada clase para la que se tenga suficientes muestras de adaptación:
     2.1. Se calcula la media y la varianza de los componentes de la clase.
     2.2. Se crean dos nuevos hijos. A cada hijo se le asigna una media igual a la del padre pero perturbada por una fracción de la varianza, en dirección contraria para cada hermano.
     2.3. Cada muestra del nodo padre se asigna al hijo con cuya media presente una distancia Euclidea más pequeña.
  3. Ir al punto 2 mientras aun hayan clases que puedan ser divididas.

- Mixta: Un experto utiliza su conocimiento sobre la tarea para decidir unas clases de regresión, y luego se expanden aquellos nodos del árbol para los que se tenga suficientes muestras.

En el presente trabajo se ha utilizado para la estimación de las clases de regresión, tanto un esquema basado en conocimiento, donde las clases de regresión se deciden siguiendo el conocimiento que se tiene sobre la tarea, como un esquema dirigido por datos, donde las clases se infieren de manera automática. Este último modo se realizado en tres fases: primero se ha construido el árbol de regresión. A continuación se ha etiquetado cada muestra de adaptación con la referencia de la gausiana que con mayor probabilidad la hubiese generado. Finalmente, a partir del árbol de regresión y de las muestras etiquetadas se obtienen las clases de regresión. Este proceso se explica con más detalle a continuación:

1. Construcción del árbol de regresión: El espacio de muestras se ha dividido utilizando un algoritmo de partición, usando la distancia euclídea como medida de disimilitud. Empezando con un nodo inicial compuesto por todos los vectores de medias de las gausianas de los modelos, se calcula la media y la varianza de todos los vectores. A continuación se crean dos nodos hijos, a los cuales se les asigna una media calculada a partir de la de su nodo padre, perturbada en sentido opuesto para cada nodo hijo por una fracción de la varianza del padre. Cada elemento del nodo padre (vectores media de las gausianas) se asigna al hijo a cuya media esté más cercano. Una vez que

todas las componentes han sido asignadas, el proceso se repite para cada nuevo nodo hasta que se obtenga un número determinado de hojas. Las hojas son denominadas *clases de regresión base* y pueden estar compuestas por gausianas individuales o por conjuntos de ellas.

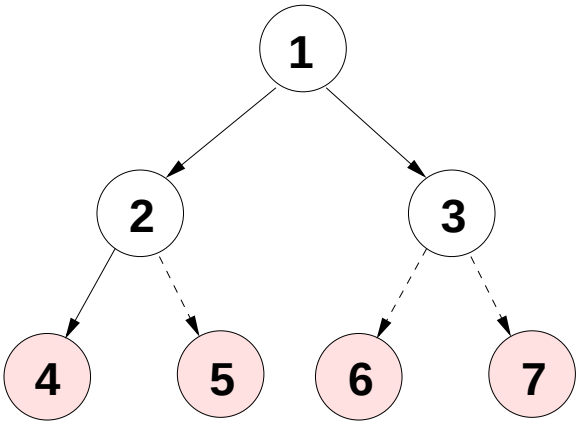


Figura 6.3: Ejemplo de árbol de regresión con cuatro clases (nodos hojas). En el proceso de adaptación, los nodos $5, 6$ y 7 no han sido visitados suficientemente (flechas discontinuas), por lo que no dispondrían de suficientes muestras para su adaptación. Por contra, el nodo 4 sí tiene suficientes muestras (flechas solidas).

2. Etiquetado de las muestras: A partir de los modelos independientes del escritor se obtiene una segmentación forzada del corpus de adaptación, esto es, el modelo de lenguaje utilizado sólo acepta la frase que se va a reconocer. Durante la segmentación se etiqueta cada muestra de adaptación con la referencia de la gausiana que con mayor probabilidad la hubiese generado.

3. Obtención de las clases de regresión: Una vez etiquetada cada muestra de adaptación, se incrementarán los contadores de los nodos del árbol de regresión que contengan el vector de medias de la gausiana con que está etiquetada cada muestra.

   En el ejemplo de la figura 6.3 se muestra un árbol de regresión con cuatro clases de regresión base $\{C_4, C_5, C_6, C_7\}$. A partir del árbol de regresión y de un conjunto de muestras de adaptación etiquetadas con referencias a gausianas, se han actualizado los contadores de los nodos. En el ejemplo de la figura 6.3 el contador del nodo 1 tendrá un valor igual al número de muestras de adaptación, puesto que contiene todas las gausianas de los modelos. El contador de muestras del estado 2 tendrá un valor igual a la suma de los contadores de los estados 4 y 5, puesto que su conjunto de gausianas es la unión del conjunto 4 y 5. En este ejemplo los contadores de las clases de regresión base C_5, C_6 y C_7 no habrían alcanzado el umbral mínimo de muestras, con lo que no se tendrían bastantes datos para su adaptación. Las gausianas

de la clase base C_5 se transformarán con el conjunto de transformaciones calculadas para el nodo interno 2, el cual si tendría suficientes muestras de adaptación. Las clases base C_6 y C_7 utilizarían el conjunto de transformaciones obtenidas para el nodo interno 3. La clase de regresión base C_4 si que tiene suficientes datos, por lo que sus transformaciones se calcularán a partir de sus propios datos. La cantidad mínima de datos necesarios para la adaptación se estima de manera empírica.

El conjunto de transformaciones a aplicar se calcula para aquellos nodos hoja para los que se tenga suficientes datos, y para aquellos nodos internos, para los que alguno de sus hijos no tenga suficientes datos, en cuyo caso, el conjunto de transformación afectará a aquellos nodos hoja (clases de regresión base) descendientes del nodo que no tengan suficientes muestras. En el ejemplo de la figura 6.3, el conjunto de transformaciones se construiría para los nodos $2, 3$ y 4. Si los conjuntos de transformaciones son $\{T_2, T_3, T_4\}$, las clases a las que afectaría cada conjunto serían:

$$\left\{ \begin{array}{lcl} \mathbf{T}_2 & \rightarrow & \{C_5\} \\ \mathbf{T}_3 & \rightarrow & \{C_6, C_7\} \\ \mathbf{T}_4 & \rightarrow & \{C_4\} \end{array} \right\} \tag{6.4}$$

Es importante reseñar que todas las gausianas que modelan un estado del HMM no tienen porqué pertenecer a una misma clase de regresión, puesto que la partición del conjunto de gausianas se hace con respecto a su proximidad con otras, en el espacio de representación.

6.2.2. Estimación de las matrices de transformación

Siguiendo la aproximación propuesta por [LW95a, LW95b], en la adaptación se modificarán solamente los vectores de medias de cada gausiana. Los vectores de medias adaptados se obtienen del siguiente modo:

$$\hat{\mu} = A\mu + b \tag{6.5}$$

donde A es una matriz de $n \times n$ y b es un vector de dimensión n (dimensión del espacio de características). Usualmente esta ecuación se reescribe como:

$$\hat{\mu}_{r_g} = W_r \, \xi_{r_g} \tag{6.6}$$

donde W_r es la matriz de transformación para la clase de regresión r, de dimensiones $n \times (n+1)$, siendo n la dimensión del espacio de características. ξ_{r_g} es el vector extendido de medias tal que $\xi_{r_g} = [\, w_r, \, \mu_{r_g 1}, \, \mu_{r_g 2}, \, \ldots \, , \mu_{r_g n} \,]^T$. El subíndice r_g hace referencia a la gausiana g de la clase de regresión r. El parámetro w_r es usado para modelar desplazamientos causados por ejemplo por una adquisición de las muestras de adaptación en diferentes condiciones que la realizada para estimar los modelos HMM independientes del escritor. La matriz W_r puede ser descompuesta como:

$$W_r = [bA] \tag{6.7}$$

donde b representa el vector de desplazamientos y A es la matriz de transformación.

Buscamos aquella matriz $\hat{W}_r$ de todas las posibles, que maximice la probabilidad de generar las muestras, O_d con los modelos adaptados Θ_{Wr} con dicha matriz.

$$\hat{W}_r = \underset{W_r}{\text{argmax}}\ P(O_d|\Theta_{Wr}) \tag{6.8}$$

La estimación de la máxima verosimilitud se obtiene con el algoritmo *Expectation-Maximization* (EM). A partir de la función auxiliar estándar (para un desarrollo más detallado ver apéndice *A* de [Chr96]),

$$\begin{aligned} Q(\Theta,\hat{\Theta}) \quad = \quad & k_1 - \tfrac{1}{2}\sum_{t=1}^{T}\sum_{r_g=1}^{R}\gamma_{r_g}(t)\Big(k_{r_g} + log(|\Sigma_{r_g}|) + \\ & + (O(t) - \bar{W}_r\xi_{r_g})^T\,\Sigma_{r_g}^{-1}\,(O(t) - \bar{W}_r\xi_{r_g})\Big) \end{aligned} \tag{6.9}$$

donde:

- k_1 es una constante que depende solamente de las probabilidades de transición del modelo HMM.
- k_{r_g} es una constante de normalización asociada a la gausiana r_g.
- T es el número de muestras de adaptación.
- R es el número de gausianas de la clase de regresión.
- $\gamma_{r_g}(t)$ representa la probabilidad de estar en el estado cuya mixtura contiene la gausiana r_g en el momento t.
- $\bar{W}_r$ es el valor estimado de los parámetros en una iteración determinada. Una vez obtenida $\bar{W}_r$ para una iteración n, esta se utiliza para modificar el vector de medias y las probabilidades de γ_{r_g}, que serán usados en la próxima iteración. Las matrices W_r se inicializan aleatoriamente.

Derivando la ecuación 6.9 con respecto a $\bar{W}_{r_g}$ e igualándola a cero se obtiene la siguiente ecuación:

$$\frac{\delta Q(\Theta,\hat{\Theta})}{\delta W_{r_g}} = 0 \ \rightarrow\ \sum_{t=1}^{T}\gamma_{r_g}(t)\left(\Sigma_{r_g}^{-1}\,(O(t) - \bar{W}_{r_g}\xi_{r_g})\,\xi_{r_g}^T\right) = 0 \tag{6.10}$$

Así la formula general para calcular $\bar{W}_r$ es:

$$\sum_{t=1}^{T}\gamma_{r_g}(t)\,\Sigma_{r_g}^{-1}\,O(t)\,\xi_{r_g}^T = \sum_{t=1}^{T}\gamma_{r_g}(t)\,\Sigma_{r_g}^{-1}\,\bar{W}_r\,\xi_{r_g}\,\xi_{r_g}^T \tag{6.11}$$

La probabilidad de estar en un estado $\gamma_{r_g}(t)$ se obtiene del proceso *forward-backward* (ver sección 2.1.2). Para obtener $\bar{W}_r$ se definen dos nuevos términos. La parte izquierda de la ecuación 6.11 no depende de la matriz de la matriz de transformación y se agrupará bajo el termino Z. El segundo termino se define del siguiente modo:

$$G = V\bar{W}_r D \tag{6.12}$$

donde,

$$V = \sum_{t=1}^{T} \gamma_{r_g}(t)\Sigma_{r_g}^{-1} \qquad\qquad D = \xi_{r_g}\xi_{r_g}^{T} \tag{6.13}$$

de este modo la eqcuación 6.11 puede reescribirse como:

$$Z = G \quad \rightarrow \quad Z = V\bar{W}_r D \tag{6.14}$$

Operando se llega a la conclusión de que $w_i^T = G_i^{-1} z_i^T$, donde w_i es el i-esimo vector de $\bar{W}_r$ y z_i es el i-esimo vector de Z (para más detalle, ver sección 5.5 de [Chr96]).

6.3. Experimentación

Las pruebas se realizaron solamente con el corpus IAMDB puesto que para el corpus ODEC no se tenían más de 10.000 muestras por escritor para realizar la adaptación. En la tabla 6.1 se muestra el número de vectores de características disponibles para adaptación y para testear los modelos adaptados.

Escritor	Vec Adap	Vec Test
000	298483	221321
548	14059	15767
551	40728	35419
552	28209	20697
584	11260	15560
588	41099	23190

Tabla 6.1: Numero de vectores de características disponibles para adaptación y para test por escritor.

		Modelos adaptados			
Escritor	Sin Adaptar	1 Cl. Reg.	2 Cl. Reg.	7 Cl. Reg.	Cl. Aut.
000	34.5	33.7	32.9	32.7	**32.3**
548	22.0	21.2	**15.9**	19.7	19.7
551	35.6	32.8	32.8	33.8	**32.5**
552	13.7	14.5	13.7	**12.6**	13.5
584	19.8	19.2	**18.6**	19.8	19.2
588	**39.6**	43.1	42.7	39.6	41.5

Tabla 6.2: Resultados por escritor para modelos sin adaptar y adaptados. Para los modelos adaptados, resultados para 1 clase de regresión, dos y siete, y número de clases estimadas a partir de los datos.

Para la primera prueba, se construyó una única clase de regresión para cada escritor con todas las gausianas de sus modelos morfológicos. La segunda prueba, consistió en dividir las gausianas en dos clases de regresión, una con las gausianas del modelo del blanco, común entre todos los escritores y que por lo tanto no dice nada sobre el estilo de escritura, además de constituir un porcentaje alto de las muestras disponibles, y otra clase para la resta de modelos. En una tercera prueba se generó un árbol manual de regresión con siete clases:

1. Modelo para el blanco.
2. Símbolos de puntuación: ( ) * + , - . / | : ; ? ! # &
3. Dígitos.
4. Mayúsculas: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
5. Caracteres minúsculos sin ascendentes ni descendentes: a c e i m n o r s u v w x z
6. Caracteres minúsculos con ascendentes: b d h f k l t
7. Caracteres minúsculos con descendentes: g j p q y

También se exploró la construcción del árbol de regresión dirigido por datos. Los resultados de dichas pruebas se muestran en la tabla 6.2. Los valores de la tabla corresponden a los valores WER obtenidos para cada escritor. Como se aprecia en la tabla, en la mayoría de los casos la adaptación al escritor mejora la prodictividad para los escritores. Sólo en el caso del escritor 588 no se mejora. Aunque este escritor presenta un gran número de muestras para adaptación, estas no son bastante representativas del test con el que se va a enfrentar.

6.4. Resumen

En este capítulo se ha estudiado la utilización de la técnica de adaptación de modelos morfológicos (HMMs) MLLR, proveniente del campo del reconocimiento automático del habla en el campo del reconocimiento automático de texto. Esta técnica ha resultado exitosa y a demostrado que mejora las tasas de reconocimiento. La adaptación resulta especialmente útil en aquellos sistemas que vayan a ser usados por un solo escritor, ya que permite ajustar un modelo general, previamente entrenado, para que capture el estilo de escritura de dicho escritor. El tipo de adaptación estudiada es estática, mientras que la construcción del árbol de regresión se ha probado para métodos basados en conocimiento, como para métodos giados por los datos.

CAPÍTULO 7

VERIFICACIÓN DE HIPÓTESIS

Los sistemas de reconocimiento de texto manuscrito no están exentos de errores. En muchos casos, no sólo interesa saber el número de errores que producirá el sistema, sino que es importante saber qué unidades de la hipótesis producida están mal reconocidas, o no se tiene garantía de que estén bien reconocidas, para así poderlas verificar y si es necesario, corregir manualmente. De esta manera se pueden construir, por ejemplo, asistentes a la transcripción, que realizarían el grueso del trabajo, dejando para el operador humano la supervisión y corrección de aquellas palabras de las cuales se tenga poca garantía de que estén bien reconocidas.

La verificación de hipótesis tiene como objetivo detectar aquellas unidades reconocidas que son susceptibles de ser errores. Para ello es necesario estimar una medida de confianza que mida el grado de fiabilidad de cada una de las unidades. En este capítulo se exploran las técnicas más usadas en la estimación de medidas de confianza, y su aplicación en el dominio del reconocimiento de texto manuscrito. La verificación de hipótesis ha sido ampliamente utilizada en reconocimiento automático del habla con resultados muy satisfactorios [San04, WSMN01, WMN99].

El capítulo se estructura de la siguiente manera: en la primera sección se exponen las técnicas de estimación de las medidas de confianza usadas en este trabajo. En la primera parte de esta sección se expone la técnica basada en grafos de palabras, mientras que en la segunda, la estimación se basa en modelos probabilísticos *naïve* Bayes. La sección tercera expone cómo se utilizan las medidas de confianza obtenidas en el apartado anterior para verificar una hipótesis. En la sección 7.3 se exponen las principales métricas utilizadas para estimar la bondad de los sistemas de verificación. En la sección 7.4 se detallan los experimentos llevados a cabo para el estudio empírico de los métodos expuestos. Por último se presenta un resumen del capítulo.

7.1. Estimación de medidas de confianza

En general, la medida de confianza se puede definir como una función que mide el grado de verosimilitud entre cada palabra proporcionada como hipótesis por el

sistema de reconocimiento y la observación.

7.1.1. Estimación basada en grafos de palabras

La siguiente aproximación fue propuesta por Wessel et al. en [WMN99]. De la observación se ha extraído que una palabra bien reconocida suele aparecer aproximadamente en el mismo intervalo de tiempo entre las hipótesis más probables. Esto ha motivado a los investigadores a utilizar listas de las N hipótesis más probables [HBPS02, Cha97] y grafos de palabras para la estimación de medidas de confianza. En la actualidad, los grafos de palabras gozan de mayor aceptación debido a que incluyen más información [KS97, San04, WMN99, WSMN01].

Un grafo de palabras $G = (Q, A, q_i, q_f, F)$ es un grafo ponderado, dirigido y acíclico, donde:

- Q es un conjunto finito de estados. Cada estado se corresponde con un estado del modelo de lenguaje que estuvo activo durante el proceso de reconocimiento, en el instante $t \in \{1, ..., T\}$ donde T es el instante final del proceso de reconocimiento. Denotaremos cada estado de Q como q^t donde q es el estado del modelos de lenguaje correspondiente y t es el instante en el que estuvo activo.
- A es un conjunto de aristas. Cada arista es una tupla $A_i = (w, u^\tau, v^t)$ de tal manera que w es una palabra, y u^τ, v^t son los estados entre los que ocurre w en el periodo de reconocimiento $[\tau, t]$.
- $q_i \in Q$ es el estado inicial.
- $q_f \in Q$ es el estado final.
- $F : A \rightarrow \Re$ es una función que asigna a cada arista (w, u^τ, v^t), la puntuación obtenida por la palabra w, durante el reconocimiento, partiendo del estado u en el momento τ, y terminando en el momento t en el estado v.

En un grafo de palabras, cualquier camino que empiece en el estado inicial y termine en el estado final corresponde a una hipótesis.

En esta aproximación, la medida de confianza de una palabra se calcula en función de todas las puntuaciones (*scores*) de las hipótesis del grafo que contienen, más o menos en el mismo intervalo de tiempo, a esa palabra [WSMN01, WMS98]. Esto supone calcular la probabilidad *a posteriori* $P(w|X)$, de cada palabra, donde X es una secuencia de vectores de características. La probabilidad *a posteriori* de una palabra, w, que ocurre entre los estados u^τ y v^t, puede calcularse sobre un grafo de palabras, simplemente sumando las probabilidades a posteriori de todas las hipótesis que pasen por aristas de la forma (w, u^τ, v^t), y normalizando por la suma de las probabilidades *a posteriori* de todas las hipótesis que contiene el grafo.

$$P((w,u^\tau,v^t) \mid X) = \frac{P((w,u^\tau,v^t),X)}{P(X)} =$$

$$\frac{1}{P(X)} \sum_{\substack{h \in G \\ (w,u^\tau,v^t) \in h}} P(h,(w,u^\tau,v^t),X) \tag{7.1}$$

La probabilidad de la secuencia de vectores de características, $P(X)$ se aproxima como la suma de las probabilidades *a posteriori* de todas las hipótesis del grafo.

$$P(X) = \sum_h P(h,X) \tag{7.2}$$

Cálculo de las probabilidades *a posteriori*

Para calcular la probabilidad *a posteriori* de una palabra puede utilizarse el muy conocido algoritmo *forward-backward*. La probabilidad *forward* $\Phi(q^t)$ de estar en un estado q en el momento t, puede ser vista como la suma de la probabilidad de todas las hipótesis parciales que llegan a dicho estado en el momento t. La probabilidad *backward* $\Psi(q^t)$ de un estado q en el momento t, es la suma de las probabilidades de todas las hipótesis parciales que salen del estado q en el momento t. La probabilidad *a posteriori* de una palabra (ecuación 7.1), puede reescribirse como:

$$P((w,u^\tau,v^t) \mid X) = \frac{\Phi(u^\tau)\, F(w,u^\tau,v^t)\, \Psi(v^t)}{\Phi(q_f)} \tag{7.3}$$

La probabilidad *forward* calculada sobre el estado final, $\Phi(q_f)$ corresponde a la suma de las probabilidades de todas las hipótesis del grafo[1]. En esta aproximación, dada una palabra w, y los instantes de tiempo τ y t entre los que ha ocurrido, el cálculo de 7.3 proporciona la medida de confianza de la palabra.

Problema de la dispersión de la probabilidad

Es bien conocido que durante el proceso de reconocimiento suelen producirse una serie de fenómenos que afectan al cálculo de las probabilidades *a posteriori*. Estos fenómenos producen lo que se denomina, *dispersión de la probabilidad*. Estos se enumeran a continuación:

a) Una misma palabra, w ocurre en intervalos de tiempo distintos, aunque solapados entre sí, lo que significa que en el grafo de palabras hay varias hipótesis posibles, con segmentaciones distintas para la palabra w. La figura 7.1 ilustra un ejemplo de este fenómeno.

[1]Esta probabilidad es igual a la probabilidad *backward* calculada sobre el primer estado, $\Phi(q_f) = \Psi(q_i)$

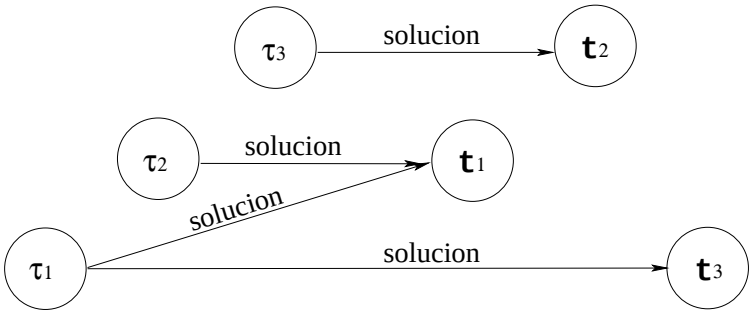


Figura 7.1: Ejemplo de grafo de palabras donde la palabra $solucion$ presenta diferentes segmentaciones.

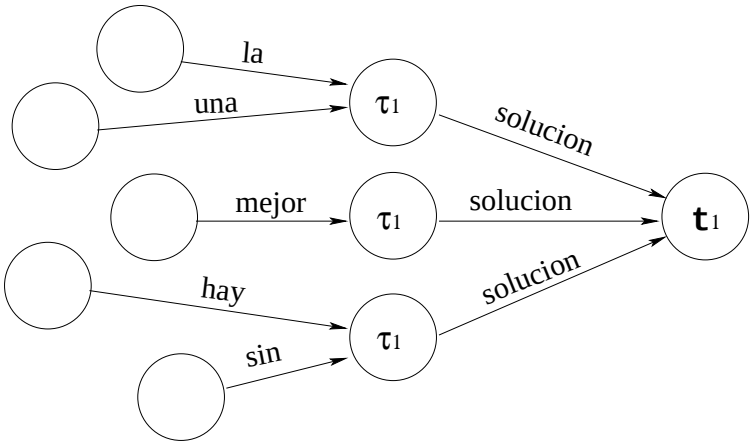


Figura 7.2: Ejemplo de palabra con la misma segmentación en el grafo de palabras, cuyas historias difieren.

b) Una misma palabra w ocurre en intervalos idénticos, pero con historias diferentes. En la figura 7.2 se muestra un ejemplo de este fenómeno.

En el cálculo de (7.3) sólo se tienen en cuenta aquellas hipótesis que contienen a la arista (w, u^τ, v^t). La estimación de la probabilidad *a posteriori*, sólo tiene en cuenta aquellas hipótesis del grafo de palabras que contienen a w, exactamente en el intervalo de tiempo (τ, t), y cuya historia coincida. Estas restricciones fuerzan a que la probabilidad *a posteriori* se disperse dependiendo de las diferentes segmentaciones que se solapen en el tiempo. Dadas todas las posibles segmentaciones, $S = \{(\tau_1, t_1), \cdots (\tau_n, t_n)\}$ en el grafo, para la palabra w, habrá dispersión de probabilidad si alguno de sus intervalos de tiempo $[\tau_i, t_i]$ se solapan, esto es, $\bigcap[\tau_i, t_i] \neq \emptyset \;\; \forall(\tau_i, t_i) \in S$, en cuyo caso, la estimación de probabilidad no será satisfactoria. En [WMS98, WSMN01] se proponen distintas soluciones a este problema. Estas soluciones pretenden acumular la probabilidad *a posteriori* de cada palabra teniendo en cuenta si hay solapamiento en el grafo de palabras.

Sea (w, u^τ, v^t) una arista del grafo de palabras y $t' \in [\tau, t]$ un instante del intervalo de tiempo en el que ocurre w, se define $P_{t'}((w, u^\tau, v^t) \mid X)$, como la suma de probabilidades *a posteriori* de las hipótesis del grafo que contienen a w en el instante t'. La estimación de la probabilidad *a posteriori* puede ser reformulada de las siguientes maneras:

- para todo $t' \in [\tau, t]$ se escoge la probabilidad *a posteriori* de los valores máximos calculados. De manera más formal:

$$P((w, u^\tau, v^t) \mid X) = \max_{t' \in [\tau, t]} P_{t'}((w, u^\tau, v^t) \mid X) \tag{7.4}$$

- para todo $t' \in [\tau, t]$ se toma el valor promedio de la probabilidad *a posteriori* de los valores calculados. De manera más formal:

$$P((w, u^\tau, v^t) \mid X) = \frac{1}{t - \tau + 1} \sum_{t' \in [\tau, t]} P_{t'}((w, u^\tau, v^t) \mid X) \tag{7.5}$$

En este trabajo se utilizará (7.4), puesto que es el método que mejores resultados a dado a los autores de [WMS98, WSMN01].

7.1.2. Estimación basada en un modelo probabilístico *Naïve* Bayes

El modelo que se presenta fue propuesto por Sanchis et al. en [San04]. En esta aproximación, el cálculo de la medida de confianza se aborda como un problema clásico de clasificación en dos clases: clase correcta e incorrecta. Cada palabra se representa por un conjunto de características, y mediante un modelo probabilístico se estima la probabilidad *a posteriori* de la clase $P(c|x, w)$ ($c = 0$ para la clase correcta, y $c = 1$ para la incorrecta), donde x será un vector de características. Utilizando la regla de Bayes:

$$P(c \mid x, w) = \frac{P(x \mid c, w)\, P(c \mid w)}{\sum_{i=0}^{1} P(x \mid i, w)\, P(i \mid w)} \tag{7.6}$$

Para el cálculo de $P(x|c, w)$ se hace la asunción *naïve* Bayes, de que las características son independientes entre sí. Este modelo, por su simplicidad, permite combinar características de distinta naturaleza, y permite construir un modelo para cada palabra del vocabulario, sin tener que recurrir a complejos métodos de aprendizaje. De la asunción de independencia entre características se obtiene:

$$P(x \mid c, w) = \prod_{d=1}^{D} P(x_d \mid c, w) \tag{7.7}$$

donde D es la dimensión del vector de características.

Estimación de las distribuciones de probabilidad del modelo

Sea $M = \{m_1, m_2, \cdots, m_n\}$ un conjunto de n muestras de entrenamiento, donde cada muestra, $m_i = (x_i, c_i, w_i)$ es una terna compuesta por una palabra w_i, el vector de características que la representa x_i, y la clase a la que pertenece, c_i. La estimación de la distribución de probabilidad $P(c \mid x, w)$, se reduce, siguiendo la ley de Bayes (ver ecuación 7.6), a estimar las distribuciones $P(c \mid w)$, para cada posible palabra, y $P(x_d \mid c, w)$, para cada combinación de palabra y clase.

Una estimación por máxima verosimilitud de $P(c \mid w)$ es:

$$\hat{P}(c \mid w) = \frac{N(c, w)}{N(w)} \tag{7.8}$$

donde $N(c, w)$ es el número de veces que la palabra w ha sido vista en la clase c, y $N(w)$ es el número de instancias de dicha palabra en las muestras de entrenamiento. Análogamente, la probabilidad $P(x_d \mid c, w)$, $d = 1, \cdots, D$, se puede aproximar por:

$$\hat{P}(x_d \mid c, w) = \frac{N(x_d, c, w)}{N(c, w)} \tag{7.9}$$

donde $N(x_d, c, w)$ es el número de veces que la característica x_d se ha visto, para la combinación de palabra y clase, y $N(c, w)$ el número de veces que se da la combinación clase y palabra.

Esta estimación exige que las características sean discretas. Por lo tanto, si la característica es de naturaleza continua es necesario discretizarla. Para cada característica continua, x_d, se define una función $f_d : \Re \rightarrow \{e_1, \cdots, e_k\}$, que asigne a cada valor de la característica el valor discreto e_i que le corresponde.

Suavizado de los modelos

Para la estimación de la distribución de probabilidad $P(c \mid x, w)$ se requiere que el conjunto de muestras de entrenamiento tenga, para cada palabra posible, un número suficientemente representativo de muestras. Este requisito es difícilmente alcanzable, sobretodo para palabras poco representadas en la muestra, y para la clase incorrecta[2]. Además, si algún evento no ha ocurrido en los datos, la probabilidad asignada por el modelo será cero. Estos inconvenientes son bien conocidos en el campo del modelado estadístico y se abordan mediante técnicas de suavizado. En este modelo se ha adoptado la técnica de suavizado *absolute discounting* [WSMN01] donde se resta masa de probabilidad a los eventos vistos, para ser repartida entre los eventos no vistos. Los detalles sobre el suavizado del modelo pueden consultarse en [San04].

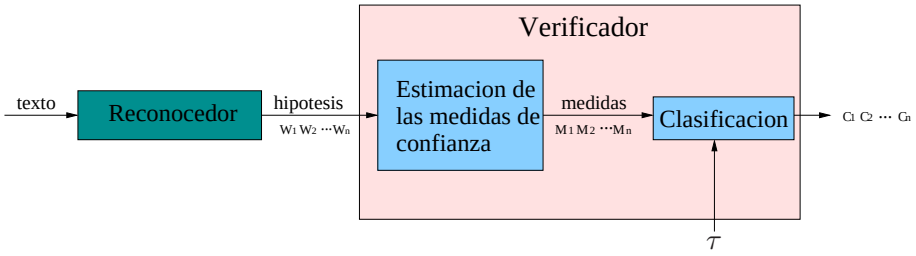


Figura 7.3: Esquema general de un sistema de verificación de hipótesis, donde w_i corresponde a la i-ésima palabra de la hipótesis, M_i es su correspondiente medida de confianza, y c_i es la clase asignada por el verificador a la palabra i.

7.2. Verificación de la hipótesis

El problema de la verificación puede verse como un problema de clasificación en dos clases: correcta e incorrecta. En la figura 7.3 se muestra un esquema general del proceso de verificación. Como todo problema de clasificación en dos clases, se puede incurrir en dos tipos de errores de clasificación: clasificar una palabra correctamente reconocida en la clase incorrecta, también conocido como *falsa alarma*, y clasificar una palabra mal reconocida en la clase correcta, también conocido como *falso positivo*. Siguiendo la teoría de decisión de Bayes [DH74], decidir que una palabra, representada por su vector de características, pertenece a la clase 0, es de mínimo riesgo si:

$$(\lambda_{10} - \lambda_{00})P(c = 0 \mid x) > (\lambda_{01} - \lambda_{11})P(c = 1 \mid x) \tag{7.10}$$

donde λ_{ij} es el perjuicio que conlleva decidir en favor de la clase i cuando la clase es la j. Sabiendo que $P(c = 1 \mid x) = 1 - P(c = 0 \mid x)$, y operando con la ecuación 7.10, el mínimo riesgo de decidir $c = 0$ se tendrá para:

$$P(c = 0 \mid x) > \frac{\lambda_{01} - \lambda_{11}}{\lambda_{10} - \lambda_{00} + \lambda_{01} - \lambda_{11}} = \tau \tag{7.11}$$

De lo que se deduce que la decisión de clasificación óptima se tomará en función de si la probabilidad de pertenecer a la clase correcta es mayor que un cierto umbral $P(c = 0 \mid x) > \tau$, que depende a su vez de los valores de λ_{ij}. En el caso que atañe, $P(c = 0 \mid x)$ es la medida de confianza calculada con alguno de los dos métodos descritos previamente.

7.3. Medidas de evaluación

Para la evaluación de los sistemas de verificación se han propuesto diferentes medidas [MH99]. A continuación se exponen las medidas más aceptadas, y las que se van a utilizar en este trabajo.

[2]Para tareas con una tasa de error de reconocimiento bajo.

7.3.1. Curvas ROC

Supóngase un sistema de reconocimiento que para una tarea determinada proporciona una serie de hipótesis con I palabras mal reconocidas y C palabras bien reconocidas. Supóngase también, que se dispone de un sistema de verificación que para un umbral dado τ, detecta D palabras de las incorrectas; $0 \leq D \leq I$ y clasifica (erróneamente) R palabras de las bien reconocidas como incorrectas; $0 \leq R \leq C$. Basándose en estos valores, se pueden definir las siguientes medidas:

1. **Tanto por uno de palabras incorrectas detectadas**,TRR (del inglés, *true rejection rate*): es el cociente entre el número de palabras mal reconocidas clasificadas como incorrectas, y el número total de palabras mal reconocidas, o lo que es lo mismo, el porcentaje (normalizado entre 0 y 1) de palabras incorrectas detectadas.

$$TRR = \frac{D}{I} \tag{7.12}$$

2. **Tanto por uno de palabras correctas rechazadas**, FRR (del inglés, *false rejection rate*): es el cociente entre el número de palabras bien reconocidas que se clasifican como incorrectas R, y el número total de palabras bien reconocidas. Dicho de otro modo, es el porcentaje (normalizado entre 0 y 1) de palabras bien reconocidas clasificadas como incorrectas.

$$FRR = \frac{R}{C} \tag{7.13}$$

Una curva ROC (Receiver Operating Characteristic) es una curva que indica para cada posible umbral $\tau \in [0, 1]$ el valor de TRR frente al valor FRR. Las curvas ROC se utilizan, no sólo para evaluar el sistema de verificación, sino que también se utilizan para elegir un punto de funcionamiento τ, que nos garantice un compromiso aceptable entre el valor de TRR y FRR.

Las curvas ROC tienen dos casos destacables: caso mejor y peor. En el caso mejor, el valor TRR es igual a uno para cada posible valor de FRR. Esto significa que el sistema de verificación clasifica todas las palabras mal reconocidas como incorrectas y las bien reconocidas como correctas. En el caso peor se cumple que $TRR = FRR$ para todo posible umbral de decisión τ. Es decir, el verificador no tiene ningún poder de discriminación. Lo habitual es encontrar curvas comprendidas entre estos dos casos (ver figura 7.4). La respuesta del sistema de verificación será tanto mejor cuanto más se acerque la curva al caso mejor.

7.3.2. AROC

El valor AROC se obtiene a partir de la curva ROC. El valor AROC se define como el área por debajo de la curva ROC dividido por el área por debajo de la curva ROC del peor caso. El valor AROC está definido en el rango $[1, 2]$, donde el

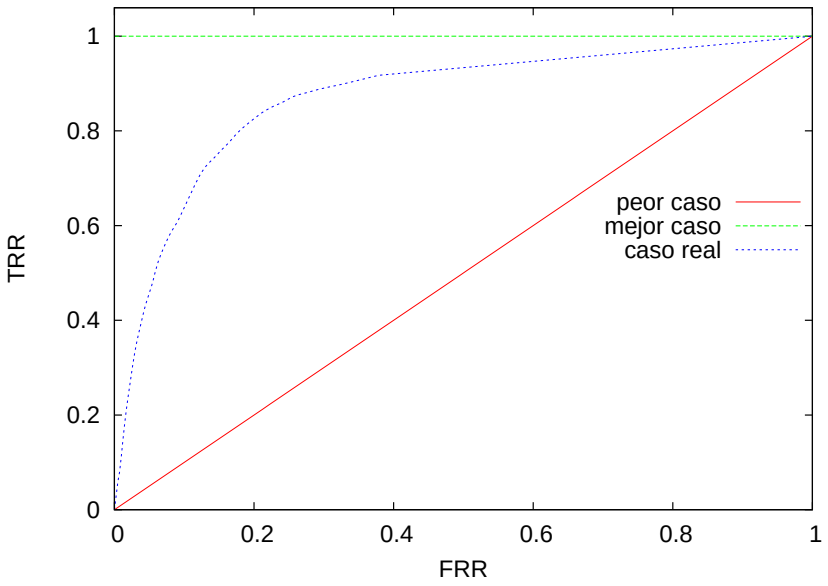


Figura 7.4: Ejemplo de curva ROC.

modelo que se ajusta al caso peor toma el valor 1, mientras que el modelo que no comete errores de clasificación toma el valor 2.

7.3.3. CER: Tasa de error de clasificación

Como ya se ha mencionado previamente, el proceso de verificación puede cometer dos tipos de errores: clasificar como incorrecta una palabra correctamente reconocida (error de tipo 1) o clasificar como correcta una palabra mal reconocida (error de tipo 2). Nótese que la cantidad de errores de tipo 1 coincide con R, y que el número de errores de tipo 2 es igual $I - D$. El CER (*classification error rate*) es el porcentaje de palabras que son clasificadas erróneamente por el sistema de verificación. El CER se define como:

$$CER = \frac{R + (I - D)}{I + C} \cdot 100 \tag{7.14}$$

Los valores R y D dependen del parámetro τ, por lo que el valor óptimo de CER se obtiene como el mínimo de los valores de CER al variar $\tau \in [0, 1]$. Los valores CER pueden calcularse directamente a partir de la curva ROC. Para cada punto de la curva se puede calcular su correspondiente valor CER ya que, $R = FRR \cdot C$ y $D = TRR \cdot I$. De esta forma, dado un punto de la curva ROC, su CER correspondiente se puede calcular como:

$$CER(FRR, TRR) = \frac{(FRR \cdot C) + (I - (TRR \cdot I))}{I + C} \cdot 100 \tag{7.15}$$

Un aspecto importante de esta medida es que permite establecer criterios relativos a la aportación del sistema de verificación al sistema de reconocimiento. Ello es debido a que se puede establecer un punto inicial (*baseline*), que será el error de clasificación del sistema de reconocimiento, a partir del cual se pueden medir las aportaciones del sistema de verificación. En el caso de un sistema al que no se le aplique ningún proceso de verificación, el número de palabras correctas rechazadas será cero ($R = 0$), a su vez el número de palabras incorrectas detectadas será cero también ($D = 0$). Su CER será:

$$CER_{baseline} = \frac{0 + (I - 0)}{I + C} \cdot 100 = \frac{I}{I + C} \cdot 100 \tag{7.16}$$

Esta situación se corresponde al caso en que el sistema de verificación clasifica todas las palabras de la hipótesis como correctas. En una curva ROC, el $CER_{basenline}$ corresponde al punto $(0, 0)$, donde ninguna palabra correcta es marcada como incorrecta y ninguna palabra incorrecta es detectada. Cualquier sistema de verificación que no mejore este valor, no estará aportando nada al sistema de reconocimiento.

7.4. Experimentación

Los grafos de palabras obtenidos a partir del proceso de reconocimiento, pueden contener básicamente tres tipos de información para cada palabra: la puntuación de los modelos morfológicos (HMMs), la probabilidad del modelo del lenguaje, y la puntuación total del sistema para esa palabra. También se tienen dos métodos para compensar la dispersión de la probabilidad: *máximo* (expresión 7.4) y *media* (expresión 7.5).

Para los experimentos donde las medidas de confianza se obtienen mediante la aproximación de grafos de palabras, se ha utilizado únicamente la información de la puntuación total del sistema, y como método de compensación de la dispersión, el *máximo*. Se ha elegido esta combinación porque es la que proporciona mejores resultados según los trabajos de [San04, WMN99, WSMN01].

Para el modelo probabilístico *naïve* Bayes, las características que representan a cada palabra se obtienen a partir del grafo de palabras. De ese modo, para cada palabra de la hipótesis se obtienen seis características: los tres tipos de puntuación del grafo por dos maneras de compensar la dispersión (ver tabla resumen 7.1).

	Método de Compensación	
Medidas	Máximo	Media
REC	RECmax	RECmed
ML	MLmax	MLmed
HMM	HMMmax	HMMmed

Tabla 7.1: Tabla de todas las características que se pueden obtener a partir del grafo de palabras. REC es la puntuación total del sistema, ML es la puntuación del modelo de lenguaje, y HMM es la puntuación de los modelos morfológicos.

Los datos de entrenamiento para el modelo probabilístico se obtienen a partir de los grafos de palabras producto de reconocer la partición de entrenamiento. Por falta de datos se utiliza la misma partición que la empleada para entrenar los HMM. Lo ideal sería entrenar a partir de una nueva partición, ya que de este modo, las muestras con las que se entrena el modelo serían independientes del escritor, tal como lo son las que se verán en test.

Característica	AROC	CER	MR (%)
RECmax	**1.69**	**17.9**	**30.6**
RECmed	1.68	18.4	28.7
MLmax	1.25	25.2	2.4
MLmed	1.25	25.2	2.4
HMMmax	1.53	24.0	7.0
HMMmed	1.51	24.8	3.9
Baseline	–	25.8	–

Tabla 7.2: Tabla resumen de los resultados para el corpus ODEC, y para cada característica, obtenidos a partir de modelos probabilísticos. MR es la mejora relativa, expresada como porcentaje, respecto al valor de *Baseline* (no hacer nada).

Resultados para el corpus ODEC

En la tabla 7.2 se presentan los resultados obtenidos con el modelo probabilístico para el corpus ODEC, utilizando sólo una característica. La característica con la que se obtiene mejor resultado es RECmax (la puntuación del sistema, más método de compensación de la dispersión *máximo*). Aunque para todos los casos la utilización del método de compensación *máximo* mejora al de *media*, la diferencia entre ambos métodos no es significativa. Donde si se aprecia gran diferencia es en la utilización de los diferentes tipos de características. El que produce mejores resultados proporciona es el basado en la puntuación del sistema, con una mejora relativa alrededor del 31 %, mucho mayor que la proporcionada por las características basadas en la puntuación del modelo del lenguaje, sólo un 2.4 %, o la

HAY MUCHA PUBLICIDAD ENGAÑOSA EJEMPLO ESTOS PUNTOS YO TENGO QUE CAMBIAR EL MOVIL Y CON PUNTOS Y TODO CUANTO ME CUESTA
HAY MUCHA PUBLICIDAD QUE ENVIAR USARIA ESTOS PUNTOS YO TENGO QUE CAMBIAR EL MOVIL Y CON PUNTOS Y TODO SOBRE CUANTO ME ENCUENTRO

INSISTO EN EL COSTE DE LAS LLAMADAS
INSISTO LLEGAN EL LA COSTE DE LAS LLAMADAS

DESEO CAMBIAR DE MOVIL A OTRO MAS MODERNO CONSERVANDO EL NUMERO ACTUAL COMO LO PODRIA HACER
DESEO CAMBIAR DE MOVILINE A OTRO MOVISTAR MODERNO CONSERVANDO EL APARTADO ACTUAL SOLO QUE LO PODRIA HACER

Tabla 7.4: Ejemplos de salida del módulo de verificación para la tarea ODEC fijando el umbral τ a 0.9855. Para cada bloque de dos frases, la primera corresponde con la transcripción correcta, mientras que la segunda corresponde a la hipótesis del sistema. En color rojo se marcan las palabras incorrectas detectadas (aciertos), en color azul las incorrectas no detectadas (errores), y en verde las correctas clasificadas como incorrectas (errores). Las palabras marcadas en negro son palabras correctas clasificadas como correctas (aciertos).

de las características basadas en la puntuación de los modelos HMMs, menos del 10 %. Como se ha comentado anteriormente, el valor mostrado para *Baseline* es el porcentaje de error del sistema si no se aplica ningún tipo de verificación.

Técnica	AROC	CER	MR (%)
Mejor Prob.	1.69	17.9	30.6
Naïve Bayes	**1.75**	**16.9**	**34.5**
Grafo palabras	1.74	17.8	31.8
Baseline	–	25.8	–

Tabla 7.3: Tabla resumen de los mejores resultados para el corpus ODEC.

En la tabla 7.3 se muestran los resultados comparativos, para el corpus ODEC de todas las técnicas de verificación pobadas. El la primera fila se presentan el mejor resultado de la tabla 7.2 para características aisladas. En la segunda fila, se muestra el resultado de verificación para modelos probabilísticos *naïve* Bayes. En *Grafo palabras* se muestran los resultados para el método de verificación basado en grafos de palabras. El mejor resultado se obtiene para la combinación de carac-

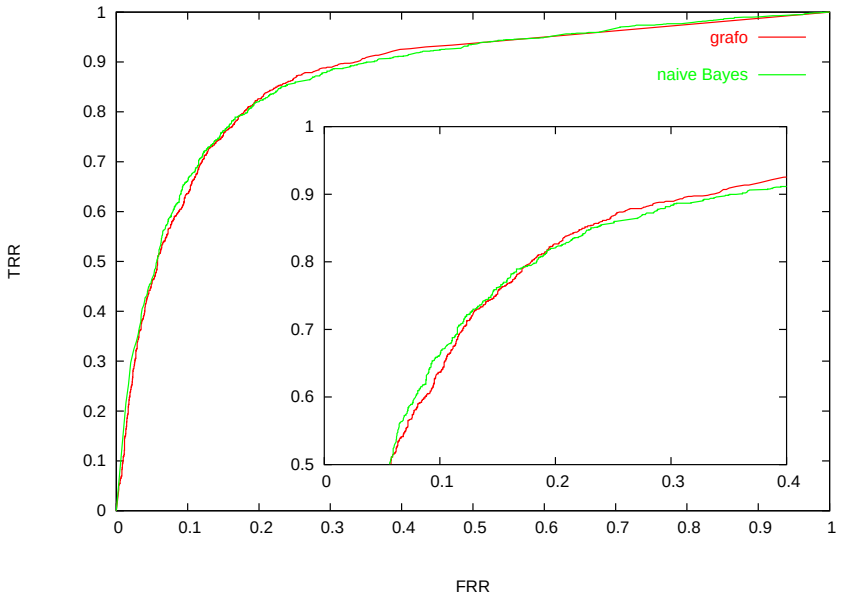


Figura 7.5: Curva ROC para ODEC mediante grafos de palabras y modelos probabilísticos *naïve* Bayes.

terísticas *naïve* Bayes con una mejora relativa del 50 %.

La figura 7.5 muestra las curvas ROC para la técnica basada en grafo de palabras y para la basada en modelos probabilísticos *naïve* Bayes. En esta curva se aprecia la poca diferencia que presentan los dos métodos.

Resultados para el corpus IAMDB

Característica	AROC	CER	MR (%)
RECmax	1.69	18.2	28.9
RECmed	1.69	18.3	28.5
MLmax	1.37	25.3	1.2
MLmed	1.39	25.1	2.0
HMMmax	1.55	22.3	12.9
HMMmed	1.53	22.6	11.7
Baseline	–	25.6	–

Tabla 7.5: Tabla resumen de los resultados para el corpus IAMDB, y para cada característica, obtenidos a partir de modelos probabilísticos. MR es la mejora relativa, expresada como porcentaje, respecto al valor de *Baseline* (no hacer nada).

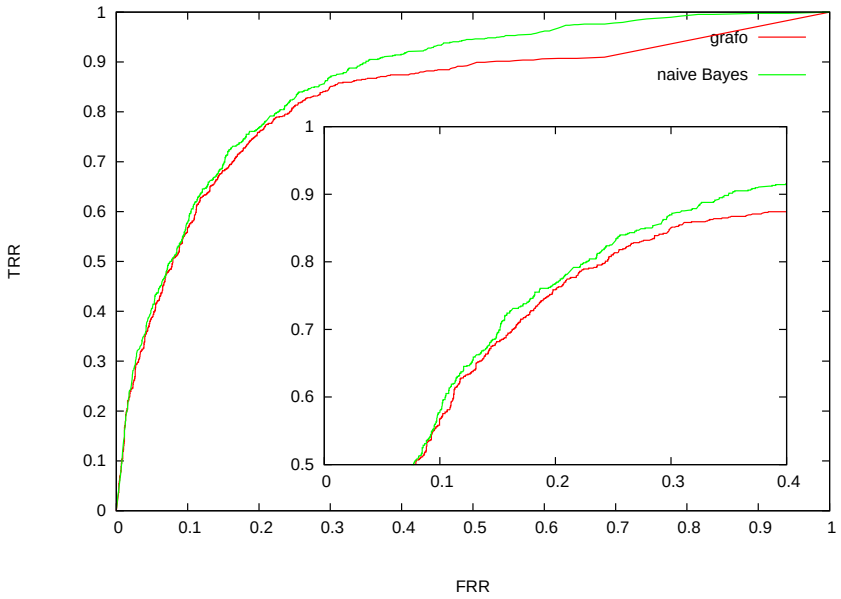


Figura 7.6: Curva ROC para el corpus IAMDB obtenida para las técnicas basadas en grafos de palabras y en modelos probabilísticos.

En la tabla 7.5 se presentan los resultados obtenidos con el modelo probabilístico para el corpus IAMDB, utilizando solamente una característica. La característica con la que se obtiene mejor resultado es RECmax (la puntuación del sistema, y con el método de compensación de la dispersión *máximo*), igual que ocuría para el corpus ODEC. En todos los casos el método de compensación de la dispersión de la probabilidad *máximo* mejora al de *media*, aunque los resultados no presentan grandes diferencias.

En la tabla 7.6 se muestran los resultados para el corpus de texto manuscrito IAMDB, con todas las técnicas de verificación probadas. En la primera fila se presenta el mejor resultado obtenido utilizando modelos probabilísticos con una sola característica. En la segunda fila se presenta el resultado para la mejor combinación de características combinandolas con el método *Naïve Bayes*. En la linea etiquetada como *Grafo de palabras* se presenta el resultado para el método basado en grafo de palabras. El mejor resultado obtenido es el producido por la técnica *Naïve Bayes*, con una mejora relativa del 31.6.

Técnica	AROC	CER	MR (%)
Mejor Prob.	1.69	18.2	28.9
Naïve Bayes	1.73	17.5	31.6
Grafo palabras	1.66	18.3	27.4
Baseline	–	25.6	–

Tabla 7.6: Tabla resumen de los mejores resultados para el corpus ODEC.

La figura 7.6 muestra las curvas ROC para la técnica basada en grafo de palabras y para la basada en modelos probabilísticos *naïve* Bayes. En esta gráfica se aprecia la poca diferencia que presentan los dos métodos, aunque el método *naïve Bayes* siempre mejora al de *grafo de palabras*.

7.5. Resumen

En este capítulo se ha estudiado la aplicación al reconocimiento de texto manuscrito *off-line* dos técnicas de verificación aplicadas con éxito en el dominio del reconocimiento automático de voz.

La primera técnica está basada en estimar la probabilidad *a posteriori* $P(w|X)$ sobre un grafos de palabras. Este método consiste en realizar una normalización de las puntuaciónes de los arcos del grafo. Para ello, se aproxima la distribución $P(X)$, por la probabilidad *forward* del último estado del grafo, $\Phi(q_f)$.

La segunda técnica utiliza un modelo probabilístico en el que se estima la probabilidad *a posteriori* de que una palabra pertenezca a cada posible clase: correcta e incorrecta. Cada palabra se representa por un conjunto de características y el modelo las combina bajo la asunción *naïve* Bayes de que son independientes entre sí. La simplicidad de estos modelos permite, por una parte, combinar fácilmente características de distinta naturaleza, y por otro lado, evita la necesidad de utilizar métodos de aprendizaje complejos.

Aunque los modelos probabilísticos *naïve* Bayes mejoran ligeramente los resultados respecto de los métodos basados en grafos de palabras, estos últimos son más baratos puesto que no requieren ningún tipo de entrenamiento, con lo que no hace falta ningún tipo de adaptación o entrenamiento en el caso de que se tengan que utilizar con nuevas tareas.

CAPÍTULO 8

CONCLUSIONES Y TRABAJOS FUTUROS

En esta tesis se han estudiado varios aspectos relacionados con la robustez de los sistemas automáticos de reconocimiento de texto (RATM), entre otros, se ha mostrado la importancia de normalizar la señal de entrada. Aunque cada sistema suele desarrollar su propia normalización, dependiendo del reconocedor, las características utilizadas, o la propia tarea, existen algunas normalizaciones estándares como la corrección del *slope*, *slant* o de la altura, por poner un ejemplo. Como consecuencia de este estudio se ha desarrollado gran cantidad de herramientas utilizadas en la actualidad por el grupo de investigación PRHLT.

Se han explotado las similitudes entre el RATM y el reconocimiento automático del habla (RAH). De esta manera se han importado dos técnicas explotadas con éxito en el campo del RAH: la adaptación de los modelos morfológicos (hmms) al locutor y la verificación de hipótesis. Ambas técnicas han probado su utilidad en el dominio del RATM. A su vez, se ha adaptado el reconocedor automático del habla ATROS para se utilizado como RATM y se ha modificado para extraer la información necesaria para realizar adaptación y para ser utilizada para verificación de hipótesis.

En un futuro póximo, y siguiendo en la misma linea de aumentar la robustez de la entrada y explotando la similitud entre el RATM y el RAH, se preve estudiar el comportamiento de modelos morfológicos contextuales. Estos modelos capturan la influencia que ejerce el contexto en la escritura de cada caracter. Un caracter no se escribe igual en cualquier contexto, sobre todo el ataque y la salida del caracter, debido a la inercia del instrumento utilizado para escribir.

En otra linea, se preve estudiar el comportamiento de modelos morfológicos emparejados (*coupled hmm*), de tal manera que se puedan combinar más de una fuente de información para el reconocimiento del texto. Esto permitirá construir sistemas de ayuda a la transcripción multimodales que permitan aprovechar la imagen de texto, y a la vez, la voz del experto en transcripción.

APÉNDICE A

EJEMPLOS DE SALIDA DEL VERIFICADOR

Umbral escogido: 0,9855; TRR: 0.5; FRR: 0.05
Significado de los colores:

- Rojo: incorrecta detectada (acierto)
- Azul: incorrecta no detectada (error)
- Verde: correcta clasificada como incorrecta (error)
- Negro: correcta clasificada como correcta (acierto)

ME GUSTARIA TENER ALGUN BENEFICIO ECONOMICO AL COMUNICARME CON OTROS TERMINALES DE MOVISTAR DENTRO DE MI MISMA FAMILIA MUJER E HIJAS
ME GUSTARIA QUE TENGO ALGUN BENEFICIO ECONOMICO AL COMUNICARME CON OTROS TERMINALES DE MOVISTAR EN PRINCIPIO DE MOVISTAR ME FAMILIA MUJER ESTA COMPAÑIA

EN LAS TARJETAS DE PREPAGO HAY ALGUNOS MOVILES A LOS QUE NO_LE FUNCIONA EL SERVICIO DE CONSULTA DE SALDO V 3 6 8 8 POR QUE MAS INFORMACION
EN LAS TARJETAS DE PREPAGO HAY ALGUNA MOVILES A LA NUEVA NO_LE FUNCIONA EL SERVICIO DE CONSULTA DE AÑO LA RECOMIENDO CON EL PROBLEMA QUE PRESTAN MAS "INFORMACION

PONER UNA O VARIAS PERSONAS EN ATENCION TELEFONICA QUE TOME DECISIONES O SE HAGA CARGO DE NUESTRA RECLAMACIONES O QUEJAS FACTURAS FALLOS
PONER UNA OFERTA VARIAS PERSONAS EN ATENCION TELEFONICA QUE TENGO MI DECISIONES OFERTA SE HAGA CARGO DE NUESTRA RECLAMACIONES OPERARIO QUEJAS FACTURAS FALLOS

INSISTO EN EL COSTE DE LAS LLAMADAS
INSISTO LLEGAN EL LA COSTE DE LAS LLAMADAS

HAY MUCHA PUBLICIDAD ENGAÑOSA EJEMPLO ESTOS PUNTOS YO TENGO QUE CAMBIAR EL MOVIL Y CON PUNTOS Y TODO CUANTO ME CUESTA
HAY MUCHA PUBLICIDAD QUE ENVIAR USARIA ESTOS PUNTOS YO TENGO QUE CAMBIAR EL MOVIL Y CON PUNTOS Y TODO SOBRE CUANTO ME ENCUENTRO

DEBERIA TENERSE MUCHA MAS ATENCION CON CLIENTES DE ANTIGÜEDAD CONSIDERABLE
DEBERA TENERSE MUCHO MAS ATENCION QUE CON CLIENTES DE ANTIGÜEDAD CONSIDERADO

MAS COBERTURA CAMBIO DE MOVIL MAS FACIL CUANDO YA ESTAS DADO DE ALTA Y BAJADA DE PRECIOS EN LAS LLAMADAS
MAS COBERTURA O CAMBIO DE MOVIL MAS FACIL CUANDO YO ESTAS OFERTAS DE ALTA CON ALTA DE PRECIOS EN LAS LLAMADO

NO SOLO EL ANTERIORMENTE CITADO
NO SOLO LAS ANTERIORMENTE DIRECCION

SIENDO EL BUZON DE VOZ GRATUITO LAS LLAMADAS REALIZADAS AL 1 2 3 PODRIA IMPLEMENTARSE EL SISTEMA DE QUE FUESE EL SISTEMA EL QUE HICIESE UNA LLAMADA PARA QUE NO HUBIESE QUE LLAMAR OBLIGATORIAMENTE
SIENDO EL BUZON DE VOZ GRATUITO LAS LLAMADAS REALIZADAS AL SERVICIO LOS SERVICIOS QUE PODRIA IMPLEMENTARSE EL SISTEMA DE QUE FUESE EL SISTEMA EL DETALLE HICIESE UNA LLAMADA PARA QUE NO ES QUE LLAMAR OBLIGATORIAMENTE

LA INFORMACION RECIBIDA EN FORMA DE MENSAJES CORTOS ES MUY UTIL Y EFICAZ
LO INFORMACION DETALLADA INFORMA DE MENSAJES CORTOS ESTOY MUY UTIL EFICAZ

YO PERSONALMENTE QUISIERA SABER SI AL TENER MI TELEFONO MOVIL YA UN TIEMPO LO PODRIA CAMBIAR SIN COSTE ALGUNO POR SER CLIENTE HABITUAL
YO PERSONALMENTE QUISIERA SABER SI CON EL TENER QUE MOVISTAR TELEFONO MOVIL YA UN TIEMPO QUE LO POSDATA DEPRISA CAMBIAR SIN COSTE ALGUNO POR SER CLIENTE MOVISTAR

OTRAS COMPAÑIAS MIMAN MAS AL USUARIO LE CAMBIAN LOS TERMINALES GRATUITOS A MI ME LO HAN NEGADO TRES VECES
OTRAS COMPAÑIAS CON MAS A NADIE LE CAMBIAN LOS TERMINALES GRATUITOS CON TELEFONIA ME LO QUE TAMBIEN NO TENER NUMERO

CUMPLE PERFECTAMENTE CON MIS NECESIDADES
CUMPLE PERFECTAMENTE CON LAS NECESIDADES

ESTOY SATISFECHO
ESTOY SATISFECHO CON SU ESPECIE

POR QUE LE DAN MAS FACILIDADES PARA ADQUIRIR UN TERMINAL NUEVO A ALGUIEN QUE NO ES CLIENTE DE MOVISTAR
ESTOY POR QUE LE DA Y MAS FACILIDADES PARA ADQUIRIR UN TERMINAL NUEVO CON ALGUIEN QUE NO ES CLIENTE DE MOVISTAR

ME PARECE BASTANTE CERCANO Y PUNTUAL
ME PARECE BASTANTE CERCANO Y PUNTUAL

LA ATENCION TELEFONICA ES EXCELENTE EN AMABILIDAD Y PESIMA EN LA RESOLUCION DE PROBLEMAS Y CALIDAD DE LA INFORMACION APORTADA
LA ATENCION TELEFONICA ES EXCELENTE EN AMABILIDAD Y PESIMA EN LA RESOLUCION DE PROBLEMAS Y CALIDAD DE LA INFORMACION APORTADA

LAS NOVEDADES Y PRESTACIONES COMUNICADAS POR MENSAJES CORTOS NO LAS VEO DE MUCHA UTILIDAD SALVO CUANDO PERMITEN AHORRO ECONOMICO EN LLAMADAS O ENVIO DE MENSAJES
LAS NOVEDADES QUE YA PRESTACIONES COMUNICADAS POR MENSAJES CORTOS NO LA VEO DE MUCHA UTILIDAD SALVO CUANDO PERMITEN AHORRO ECONOMICO EN LLAMADAS A ENVIO DE MENSAJES

MEJORAR LA COBERTURA EN LA GOMERA CANARIAS
MEJORAR LA COBERTURA EN LA GOMERA GRACIAS

PRECIOS EXCESIVOS CON RESPECTO A OTROS PAISES
PRECIOS EXCESIVOS INTERESA INFORMACION SOBRE EL PRECIO A OTROS DAIS EXCESIVO

LA MEJOR MOVISTAR POR COBERTURA
CAMBIO A MOVISTAR A LA COBERTURA

MAYOR COBERTURA EN LAS COMUNICACIONES Y MENOS INTERFERENCIAS
MAYOR COBERTURA EN LLAMADAS POR DELANTE EN LAS COMUNICACIONES Y ANTIGUOS INTERFERENCIAS

DEBERIAN HACER PAQUETES INDIVIDUALES Y PERSONALIZADOS PARA LOS CLIENTES INTERESADOS COMO LAS HIPOTECAS BANCARIAS
DEBERIAN SATISFACER PAQUETES INDIVIDUALES COMUNICACIONES Y PERSONALIZADOS QUE PARA LOS CLIENTES INTERESADOS COMO LLAMADAS HIPOTECAS BANCARIAS

OFRECER MOVILES DE ALTA CALIDAD A UN COSTO MAS ASEQUIBLE
INFORMAR MOVILES DE ALTA CALIDAD POR UNO CON MAS ASEQUIBLE

ESTOY CONTENTO CON EL SERVICIO PRESTADO
ESTOY CONTENTO CON OTRO POR SERVICIO PRESTADO

MEJORAR SERVICIO GENERAL COBERTURA CALIDAD SONIDO ATENCION AL CLIENTE LA UNICA VEZ QUE HE LLAMADO NO CONVENCE EXPLICACION BAJAR TARIFAS
ALGUN SERVICIO GENERAL COBERTURA CALIDAD SONIDO ATENCION AL CLIENTE LA LINEA ESTE QUE HE LLAMADO UN CLIENTE DEBERIA BAJAR TARIFAS

ENVIAN LOS MENSAJES DE INFORMACION A HORAS INTEMPESTIVAS COMO LAS 5 A M
ENVIAN LOS MENSAJES DE INFORMACION A HORAS INTEMPESTIVAS COMO LLAMADAS SERIA MEJOR

DESEO CAMBIAR DE MOVIL A OTRO MAS MODERNO CONSERVANDO EL NUMERO ACTUAL COMO LO PODRIA HACER
DESEO CAMBIAR DE MOVILINE A OTRO MOVISTAR MODERNO CONSERVANDO EL APARTADO ACTUAL SOLO QUE LO PODRIA HACER

QUE SE ENTIENDAN MEJOR
QUE SE ENTIENDAN NUMERO

NO ESTARIA MAL LA EDICION DE ALGUN TIPO DE REVISTA
NO ESTARIA CON LA POSICION NI ALGUN TIPO DE MENOS

CONSULTA DE LA TARIFICACION POR INTERNET Y CAMBIOS DE TARIFA
ME CONSULTA DE LA TARIFICACION PROMOCION INTERNET Y CAMBIOS DE TELEFONO

ME REMITO A LO EXPUESTO EN EL PUNTO 6
ME REMITO A QUE LO EXPUESTO EN EL PUNTO SERVICIOS

AL PRINCIPIO MUCHAS FACILIDADES AHORA MES AL SER YA CLIENTE FIJO
AUMENTAR MEDINA FACILIDADES PARA SATISFACER AL CLIENTE FACTURACION

ÍNDICE GENERAL DE TABLAS

ÍNDICE GENERAL DE FIGURAS

BIBLIOGRAFÍA

[aEKL00] D.P. D'Amato adn E.j. Kuebert and A. Lawson. Results from a performance evaluation of handwritten address recognition systems for the united states postal service. In *7th International Workshop On Frontiers in Handwriting Recognition*, pages 189–198, Amsterdam, Holand, September 2000.

[AF00] A. Amin and S. Fischer. A document skew detection method using the hough transform. *Pattern Analysis and Applications*, 3:243–253, 2000.

[Ari98] Nafiz Arica. *An off-line character recognition system for free style handwriting*. PhD thesis, School of Natural and Applied Sciences, The Graduate Scool of Natural and Applied Sciences of The Middle East Technical University, September 1998.

[AV00] N. Arica and F.T. Yarman Vural. One dimensional representation of two dimensional information for hmm based handwritten recognition. *Pattern Recognition Letters*, 21((6-7)):583–592, 2000.

[Bai87] H.S. Baird. The skew angle of printed documents. In *Proc. of the Conf. Society of Photographic Scientists and Engineers*, pages 14–21, 1987.

[BC03] U. Bhattacharya and B.B. Chaudhuri. A majority voting scheme for multiresolution recognition of handprinted numerals. In *In 7th international Conference on Document Analysis and Recognition*, pages 16–20, 2003.

[Ber86] J. Bernesen. Dynamic thresholding of grey-level images. In *Conferece on Pattern Recognition*, pages 1251–1255, 1986.

[BRKR00] A. Brakensiek, A. Rottland, J. Kosmala, and G. Rigoll. Off-line handwriting recognition using various hybrid modeling techniques and character n-grams. In *7th International Workshop On Frontiers in Handwriting Recognition*, pages 343–352, Amsterdam, Holand, September 2000.

[BRST95] H. Bunke, M. Roth, and E.G. Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden markov models. *Pattern Recognition*, 28(9):1399–1413, 1995.

[BS89] R. Bozinovic and S. Srihari. Off-line cursive script word recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(1):68–83, 1989.

[BSM98] I. Bazzi, R. Schwartz, and J. Makhoul. An omnifont open-vocabulary ocr system for english and arabic. *Pattern Recognition*, 21(6):495–504, junio 1998.

[Bun03] Host Bunke. Recognition of cursive roman handwriting- past, present and future. In IEEE, editor, *Proc. of the 7th International Conference Document Analysis and Recognition (ICDAR '03)*, volume 1, pages 448–459, 2003.

[CC98] R. Cattoni and T. Coianiz. Geometric layout analisis techniques for document image understanding: a review. Technical report, ITC-IRST, Via Sommarive, I-38050 Povo, Trento, Italy, 1998.

[Cha97] L. Chase. *Error-responsive feedback mechanisms for speech recognizers*. PhD thesis, Scool of Computer Science, Carnegie Mellon University, USA, 1997.

[Chr96] Heidi Christensen. *Speaker Adaptation of Hidden Markov Models Using Maximum Likelihood Linear Regression*. PhD thesis, Institute of Electronic Systems. Department of Communication Technology, Aalborg (Denmark), June 1996. Advisor(s): Ove Andersen and Børge Lindberg.

[CHS94] E. Cohen, J.J. Hull, and S.N. Srihari. Control structure for interpreting handwritten addresses. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(10), 1994.

[CKS95] M.Y. Chen, A. Kundu, and S.N Srihari. Variable duration hidden markov model and morphological segmentation for handwritten word recognition. *IEEE Trans. on Image Processing*, 4(12):1675–1688, 1995.

[CKZ94] M.Y. Chen, A. Kundu, and J. Zhou. Off-line handwritten word recognition using a hidden markov model type stochastic network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5):481–496, 1994.

[CL96] R. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:690–706, 1996.

[CWL03] Yang Cao, Shuhua Wang, and Heng Li. Skew detection and correction in document images based on straight-line fitting. *Pattern Recognition Letters*, 24:1871–1879, 2003.

[CY97] K. Chung and J. Yoon. Performance comparison of several feature selection methods based on node pruning in handwritten character recognition. In *in Proc. 4th Int. Conf. Document Analysis and Recognition*, pages 11–15, 1997.

[Dau90] I. Daubechies. The wavelet transform, time frequency localisation and signal analysis. *IEEE Trans. on Information Theory*, 36(5):961–1004, 1990.

[DCB+99] V. Digalakis, H. Collier, S. Berkowitz, A. Corduneanu, E. Bocchieri, A. Kannan, C. Boulis, S. Khudanpur, W. Byrne, and A. Sankar. Rapid speech recognizer adaptation to new speakers. In *on proccedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 2, pages 765–768, 1999.

[DH74] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1974.

[DK97] M. Dar and F. Kimura. *Handbook of Character Recognition and Document Image Analisis*, chapter Segmentation-based cursive handwriting recognition, pages 123–156. H.Bunke and P.Wang, 1997.

[DRN95] V.V. Digalakis, D. Rtischev, and L.G. Neumeyer. Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE Trans. Speech and Audio Processing*, 3(5):357–366, 1995.

[eA94] Geist et Al. The second census optical character recognition system conference. Technical report, Inst. of Standards and Technology, Gaithersburg, MD 20899, 1994.

[Ell87] Douglas F. Elliott. *Handbook of Digital Image Processing Engineering Applications*. Accademic Press Inc., 1987.

[ETM91] L. Eikvil, T. Taxt, and K. Moen. A fast adaptative method for binarization of document images. In *First International Conference on Document Analysis and Recognition*, pages 435–443, 1991.

[EYGSS99] A. El-Yacoubi, M Gilloux, R. Sabourin, and C.Y. Suen. An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):752–760, Agosto 1999.

[F'u99] Keinosuke F'ukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 525 B Streeet, Sutie 1900,San Diego, CA 92101-4495, USA, 1999.

[GAA+99] N. Gorski, V. Anisimov, E. Augustin, D. Price, and J.C. Simon. A2ia check reader: A family of bank check recognition systems. In *In Proc. of 5th International Conference on Document Analysis and Recognition*, pages 523–526, 1999.

[GAA+01] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximor. Industrial ank check processing: the a2ia check reader. *International Journal on Document Analysis and Recognition*, 3:196–206, 2001.

[Gab46] D Gabor. Theory of communication. *Journal of Institute for Electrical Engineering*, 93(III):429–457, 1946.

[Gal96] M.J Gales. The generation and use of regression class trees for mllr adaptation. Technical report, Cambridge University Engineering Department, 1996.

[Gal00] M.J.F. Gales. Cluster adaptive training of hidden markov models. *IEEE Trans. on Speech and Audio Processing*, 8:417–428, 2000.

[Gar92] M.D. Garris. Design and collection of a handwritten sample image database. social science computer review, volume 10. Technical report, Inst. of Standards and Technology, Gaithersburg, MD 20899, 1992.

[GB03] S. Guenter and H. Bunke. Optimizing the number of states training iterations, gaussians in an hmm-based handwritten word recognizer. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 472–276, 2003.

[GB04] Simon G unter and Host Bunke. Hmm-based handwritten word recognition: on the optimization of the number of states, training iterations and gaussian components. *Pattern Recognition*, 37:2069–2079, 2004.

[Gha01] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42, 2001.

[GJ92] M.D. Garris and S.A. Janet. Nist scoring package user's guide release 1. Technical report, Inst. of Standards and Technology, Gaithersburg, MD 20899, 1992.

[GL94] J.L. Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture of marcov chains. *IEEE Trans. on Speech and Audio Processing*, 2:291–298, 1994.

[GS95] D. Guillevic and C.Y. Suen. Cursive script recognition applied to the processing of bank cheques. In *Proc. Second International Conference on Document Analysis and Recognition*, pages 11–14, 1995.

[GS97] D. Guillevic and C.Y. Suen. Hmm word recognition engine. In *Proc. Third International Conference on Document Analysis and Recognition*, pages 544–547, 1997.

[GST+00] J. González, I. Salvador, A. H. Toselli, A. Juan, E. Vidal, and F. Casacuberta. Off-line recognition of syntax-constrained cursive handwritten text. In *Proc. of Joint IAPR Int. Workshops SSPR 2000 and SPR 2000*, volume 1876 of *Lecture Notes in Computer Science*, pages 143–153, Alacant (Spain), September 2000. Springer-Verlag.

[GW96] M.J.F. Gales and P.C. Woodland. Mean and variance adaptation within the mllr framework. *Computer Speech and Language*, 10:249–264, 1996.

[GWJ+94] J. Geist, R.A. Wilkinson, S. Janet, P.J. Grother, B. Hammond, N.W. Larsen, R.M. Klear, M.J. Matsko, C.J.C. Burges, R. Creecy, J.J. Hull, T.P. Vogl, and C.L. Wilson. The second census optical character recognition systems conference. Technical report, National Institute of Standards and Technology, may 1994.

[HAJ90] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburg University Press, 1990.

[HB04] Luke A. D Hutchison and William A. Barrett. Fast registration of tabular document images using the fourier-mellin transform. In *In Proc. of the First International Conference on Document Image Analysis for Libraries*, pages 253– 267, 2004.

[HBPS02] T.J. Hazen, T. Burianek, J. Polifroni, and S.Seneff. Recognition confidence scoring for use in speech understanding systems. *Computer Speech and Language*, 16(1):49–67, 2002.

[HDM+94] J.J. Hull, A.C. Downton, M.Garris, C.Y. Suen, and K. Yamamoto. Databases of off-line handwritten english text. Technical report, IAPR TC 11, 1994.

[HFD90] S. Hinds, J. Fisher, and D. D. D'Amato. A document skew detection method using runlength encoding and the hough transform. In *In Procedings of the International Conference on Pattern Recognition*, pages 464–468, 1990.

[Hu62] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8:179–187, 1962.

[Hul98] Jonathan J. Hull. Document image skew detection: survey and annotated bibliographi. *Hull, J.J, Taylor, S.L. (Eds.), Document Analysis Systems II. Word Scientific*, pages 40–64, 1998.

[HYR86] A. Hashizume, P.S. Yeh, and A. Rosenfeld. A method of detecting the orientation of aligned components. *Pattern Recognition Letters*, 4:125–132, 1986.

[IOO91] S. Impedovo, L. Ottaviano, and S. Occhiegro. Optical character recognition-a survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1):1–24, 1991.

[JBWK99] Xiaoyi Jiang, Horst Bunke, and Dubravka Widmer-Kljajo. Skew detection of document images by focused nearest-neighbor clustering. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, 1999.

[Jel98] F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts London, England, 1998.

[JLG78] S. Johansson, G. Leech, and H. Goodluck. Manual of information to accompany the lancaster-oslo/bergen corpus of british english, for use with digital computers. Technical report, Department of English, University of Oslo, 1978.

[JLN03] S. Jaeger, C.-L. Liu, and MÑakagawa. The state of the art in japanese online handwriting recognition compared to techniques in western handwriting recognition. *I: Journal on Document Analysis and Recognition*, 6:75–88, 2003.

[KABP98] S. Knerr, E. Augustin, O. Baret, and D. Price. Hidden markov model based word recognition and its application to legal amount reading on french checks. *Computer Vision and Understanding*, 70(3):404–419, junio 1998.

[Kat87] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35:400–401, 1987.

[KB00] G. Kaufmann and H. Bunke. Automated reading of cheque amounts. *Pattern Analysis and Applications*, 3:132–141, 2000.

[KB06] Douglas J. Kennard and William A. Barrett. Separating lines of text in free-form handwritten historical documents. In *In Proc. of the Second International Conference on Document Image Analysis for Libraries*, 2006.

[KDFK03] E. Kavallieratou, N. Dromazou, N. Fakotakis, and G. Kokkinakis. An integrated system for handwritten document image processing. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(4):617–636, 2003.

[KFK02] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. An unconstrained handwriting recognition system. *International Journal on Document Analysis and Recognition*, 4:226–242, 2002.

[KG97] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real time application. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(9):366–379, 1997.

[KGS99] G. Kim, V. Govindaraju, and S. Srihari. An architecture for handwritten text recognition systems. *International Journal on Document Analysis and Recognition*, 2(1):37,44, July 1999.

[KK98] H.Y. Kim and J.H. Kim. Handwritten korean character recognition based on hierarchical random graph modeling. In *In Proc. of International Workshop on Frontiers in Handwriting Recognition*, pages 577–586, 1998.

[KNSV93] M. Krishnamoorthy, G.Ñagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(7):737–747, 1993.

[Kor97] A. Kornai. An experimental hmm-based postal ocr system. *In porc of International Conference on Acoustics, Speech and Signal Processing*, 4:3177–3180, 1997.

[KS97] T. Kemp and T. Schaaf. Estimating confidence using word lattices. In *European Conf. on Speech Technology (EuroSpeech)*, pages 827–830, 1997.

[KSFK03] E. Kavallieratou, K. Sgarbas, N. Fakotakis, and G. Kokkinakis. Handwritten word recognition based on structural characteristics and lexical suport. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 562–566, 2003.

[KSS03a] A.L. Koerich, R. Sabourin, and C.Y. Suen. Large vocabulary off-line handwriting recognition. *Pattern Analysis and Applications*, 6:97–121, 2003.

[KSS03b] A.L. Koerich, R. Sabourin, and C.Y. Suen. Lexicon-driven hmm decoding for large vocabulary handwriting recognition with multiple character models. *International Journal on Document Analysis and Recognition*, 6:126–144, 2003.

[LDG+00] V. Di Lecce, A. Dimauro, Guerriero, S. Impedovo, G. Pirlo, and A. Salzo. A new hybrid approach for legal amount recognition. In *In Proc. of International Workshop on Frontiers in Handwriting Recognition*, pages 199–208, 2000.

[Lee89] K. F. Lee. *Automatic Speech Recognition: the Development of the SPHINX System*. Kluwer Academic Pub., 1989.

[Lev66] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sovietphys. Dokl.*, 10(8):707–710, 1966.

[LFK01] N. Liolios, N. Fakotakis, and G. Kokkinakis. Improved document skew detection based on text line conected component clustering. In *Proc. International converence on Image Processing*, volume 1, pages 1098–1101, Thessaloniki, Greece, 2001.

[LFK02] N. Liolios, N. Fakotakis, and G. Kokkinakis. On the generalization of the form identification and skew detection problem. *Pattern Recognition*, 35(1):253–264, 2002.

[LG93] C.H. Lee and J.L. Gauvain. Speaker adaptation based on map estimation of hmm parameters. In *In Proceedings of IEEE Conference on Audio Speech and Signal Processing*, volume 2, pages 558–561, 1993.

[LK95] S.W. Lee and Y.J. Kim. Multiresolutional recognition of handwritten numerals with wavelet transform and multilayer cluster neural network. In *in Proc. 3rd Int. Conf. Document Analysis and Recognition*, pages 1010–1014, 1995.

[LNSF02] C.-L. Liu, K.Ñakashima, H. Sako, and H. Fujisawa. Handwritten digit recognition using state-of-the-art techniques. In *In Proc. of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 320–325, 2002.

[LOHG97] X. Li, W. Oh, J. Hong, and W. Gao. Recognizing components of handwritten characters by attributed relational graphs with stable features. In *in Proc. 4th Int. Conf. Document Analysis and Recognition*, pages 616–620, 1997.

[LRS91] W. Lu, Y. Ren, and C.Y. Suen. Hierarchical attributed graph representation and recognition of handwritten chinese characters. *Pattern Recognition*, 24(7):617–632, 1991.

[LS96] Y. Lu and M. Shridar. Character segmentation in handwritten words-an overview. *Pattern Recognition*, 29(1):77–96, 1996.

[LT03] Yue Lu and Chew Lim Tan. A nearest-neighbor chain based approach to skew estimation in document imaages. *Pattern Recognition Letters*, 24:2315–2323, 2003.

[LW95a] C.J. Leggetter and P.C. Woodland. Flexible speaker adaptation for large vocabulary speech recognition. In *In Proc. of 4th European conference on Speech Communication and Technology*, pages 1155–1158, 1995.

[LW95b] C.J. Leggetter and P.C Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hmms. *Computer Speech and Language*, 9(2):171–185, 1995.

[MB99] U. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *In Proc. of the 5th Int. Conf. on Document Analysis and Recognition*, pages 705–708, 1999.

[MB00] U. Marti and H. Bunke. Handwritten sentence recognition. In *In Proc. of the 15th Int. Conf. on Pattern Recognition*, pages 467–470, 2000.

[MB01] U. Marti and H. Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.

[MB02] U. Marti and H. Bunke. The iam-database: an english sentence database for off-line handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.

[MFB+99] M. Morita, J. Facon, F. Bortolozzi, S. Garnes, and R. Saboruin. Mathematial morphology and weighted least squares to correct handwriting baseline skew. In IEEE, editor, *Proc. of the 5th International Conference Document Analysis and Recognition (ICDAR '99)*, pages 430–433, 1999.

[MG96] M. Mohamed and P. Gader. Handwritten word recognition using egmentation-free hidden markov modeling and segmentation-based dynamic programming techniques. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):548–554, 1996.

[MG99] S. Madhvanath and V. Govindaraju. Local reference lines for handwritten phrase recognition. *Pattern Recognition*, 32:2021–2028, 1999.

[MG01] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in hadwritten word recogniton. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):149–164, 2001.

[MH99] M.Siu and H.Gish. Evaluation of word confidence for speech recognition systems. *Computer Speech and Language*, 13(4):299–318, 1999.

[MKG99] S. Madhvanath, G. Kim, and V. Govindaraju. Chaincode contour processing for handwritten word recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(9):928–932, 1999.

[MM94] K.M. Mohiuddin and J. Mao. A comparative study of different classifiers for handprinted character recognition. *Pattern Recognition*, pages 437–448, 1994.

[Mor91] J. Moreau. A new system for automatic reading of postal checks. In *In Proc. of International Workshop on Frontiers in Handwriting Recognition*, pages 121–132, 1991.

[MSBS03] M.E. Morita, R. Sabourin, F. Bortolozzi, and C.Y. Suen. A recognition and verification strategy for handwritten word recognition. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 482–486, 2003.

[MSLB98] J. Makhoul, R. Schwartz, C. Lapre, and I. Bazzi. A script-independent methodology for optical character recognition. *Pattern Recognition*, 31(9):1285–1294, 1998.

[MSY92] Shunji Mori, Ching C.Y. Suen, and Kazuhiko Yamamoto. Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.

[Nag00] George Nagy. Twenty years of document image analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.

[NEK94] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependencies in stochastic language modeling. *Computer Speech and Language*, 8(1):1–28, 1994.

[Nib86] W.Ñiblack. *An Introduction to Digital Image Processing*. Englewood Cliffs, Prentice Hall, 1986.

[NT96] M.Ñakagawa and L.V. Tu. Structural learning of character patterns for on-line recognition of handwritten japanese characters. In *In Proc. of the 6th international workshop in Advances in structural and syntactical pattern recognition*, pages 180–188, 1996.

[ØDT96] T. Taxt Ø. D. Trier, A. K. Jain. Feature extraction method for character recognition - a survey. *Pattern Recognition*, 29(4):641–6628, 1996.

[Ots79] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9(1):62–66, March 1979. minimize inter class variance.

[PP02] J.F. Pitrelli and M.P. Perrone. Confidence modeling for verification post-processing for handwriting recognition. In *In Proc. of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 30–35, 2002.

[PP03] J.F. Pitrelli and M.P. Perrone. Confidence-scoring postprocessing for off-line handwritten-character recognition verification. In *In 7th international Conference on Document Analysis and Recognition*, pages 278–292, 2003.

[PS00] R. Plamondon and S.N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 22(1):63–84, 2000.

[PSCV01] M. Pastor, A. Sanchis, F. Casacuberta, and E. Vidal. Eutrans: a speech-to-speech tranlator prototype. In *In Proc. of European conference on Speech Communication and Technology*, 2001.

[PTRV06] M. Pastor, A.H. Toselli, V. Romero, and E. Vidal. Improving handwritten off-line text slant. In *Proc. of Sixth IASTED International Conference of Visualization, Imaging and Image Processing*. Iasted, August 2006.

[PTV04] M. Pastor, A.H. Toselli, and E. Vidal. Projection profile based algorithm for slant removal. *Lecture Notes in Computer Science*, 3212:183,171, September 2004.

[RJ93] L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1993.

[RK98] Gerhard Rigoll and Andreas Kosmala. A systematic comparison between on-line and off-line methods for signature verification with hidden markov models. In *Int. Conference on Pattern Recognition (ICPR)*, pages 1755–1757, Brisbane, 1998.

[Rom06] Verónica Romero. Mejora de la normalización de tamaño de texto manuscrito off-line. Master's thesis, Facultat d'Informàtica de València, 2006. Advisor(s): Moisés Pastor.

[RPTV06] V. Romero, M. Pastor, A.H. Toselli, and E. Vidal. Criteria for handwritten off-line text size normalization. In *Proc. of Sixth IASTED International Conference of Visualization, Imaging and Image Processing*. Iasted, August 2006.

[San98] A. Sankar. Experiments with a gaussian merging-splitting algorithm for hmm training for speech recognition. In *In Proc. of 1997 DARPA Broadcast News Transcription and Understanding Workshop*, pages 99–104, 1998.

[San04] Alberto Sanchis. *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*. PhD thesis, Facultat d'Informàtica, Universitat Politècnica de València, Valencia, 2004.

[Say73] K.M. Sayre. Machine recognition of handwritten words: A project report. *Pattern Recognition*, 5(3):213–228, 1973.

[SB93] R.K. Srihari and C.M. Baltus. Incorporating syntactic constraints in recognizing handwritten sentences. In *In Proc. of the International Joint Conference on Artificial Intelligence*, pages 1262–1267, 1993.

[Sch03a] M.P. Schambach. Determination of the number of writing variants with an hmm based cursive word recognition system. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 119–123, 2003.

[Sch03b] M.P. Schambach. Model length adaptation of an hmm based cursive word recognition system. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 109–113, 2003.

[Shu96] Han Shu. A on-line handwriting recognition using hidden markov models. Master's thesis, Massachusetts Institute of Technology, 1996.

[SK83] D. Sankoff and J.B. Kruskal. *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.

[SLB91] J.C. Salome, M. Leroux, and J. Badard. Recognition of cursive script words in a small lexicon. In *Proc. of the first International Conference Document Analysis and Recognition (ICDAR '91)*, pages 774–782, 1991.

[SLG+96] C.Y. Suen, L. Lam, D. Guillevic, N.W. Strathy, M. Cheriet, J.N. Said, , and R. Fan. Bank check processing system. *J. Imaging Systems and Technology*, 7:392–403, 1996.

[SP00] J. Sauvola and M. Pietik ainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[SPJ97] A. Simon, J.C. Pret, and A.P. Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(3):273–277, 1997.

[SR98] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):309–321, March 1998.

[SRI99] T. Steinherz, E. Rivlin, and N. Intrator. Off-line cursive script word recognition-a survey. *International Journal on Document Analysis and Recognition*, 2:90–110, September 1999.

[Sri00] S.N. Srihari. Handwritten address interpretation: A task of many pattern recognition problems. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(5):663–674, agosto 2000.

[SS97] Changming Sun and Deyi Si. Skew and slant correction for document images using gradient direction. In *Proceedings of the 3th International Conference on Document Analysis and Recognition*, pages 142–146, 1997.

[Ste85] F. M Stentiford. Automatic feature design for optical character recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(3):349–355, 1985.

[SWN98] T. Shioyama, H.Y. Wu, and T.Ñojima. Recognition algorithm based on wavelet transform for handprinted chinese characters. In *in Proc. 14th Int. Conf. Pattern Recognition*, pages 229–232, 1998.

[TA92] S. Tsujimoto and H. Asada. Major components of a complete text reading system. *In Procedings of the IEEE*, 80(7):1133–1149, 1992.

[Tak99] M. Takano. Unified state/mixture topology design via multi-path hmm by ml greedy state split. In *In Proc. of International Workshop on Automatic Speech Recognition and Understanding*, pages 95–98, 1999.

[TFJ89] T. Taxt, P.J. Flynn, and A.K. Jain. Segmentation of document images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1322–1329, 1989.

[TJ95] Øivind D. Trier and Anil K. Jain. Goal-directed evaluationof binarization methods. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(12):1191–1201, 1995.

[TJG+04] A. H. Toselli, A. Juan, J. González, I. Salvador, E. Vidal, F. Casacuberta, D. Keysers, and H.Ñey. Integrated handwriting recognition and interpretation using finite-state models. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(4):519–539, 2004.

[TJV04] A. H. Toselli, A. Juan, and E. Vidal. Spontaneous handwriting recognition and classification. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 433–436, Cambridge, United Kingdom, August 2004.

[TK03] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 525 B Streeet, Sutie 1900,San Diego, CA 92101-4495, USA, 2003.

[Tos04] Alejandro Héctor Toselli. *Reconocimiento de Texto Manuscrito Continuo*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia (Spain), March 2004. Advisor(s): Dr. E. Vidal and Dr. A. Juan (in spanish).

[TPJV05] A.H. Toselli, M. Pastor, A. Juan, and E. Vidal. Spontaneous handwriting text recognition and classification using finite-state models. *Lecture Notes in Computer Science*, 3523:363–371, June 2005.

[TT99] Y. Tao and Y.Y. Tang. The feature extraction of chinese character based on contour information. In *in Proc. of 5th International Conference on Document Analysis and Recognition*, pages 637–640, 1999.

[US02] S. Uchida and H. Sakoe. A handwritten character recognition method based on unconstrained elastic matching and eigen-deformations. In *In Proc. of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 72–77, 2002.

[US03] S. Uchida and H. Sakoe. Handwritten character recognition using elastic matching based on a class-dependent deformation model. In *In 7th international Conference on Document Analysis and Recognition*, pages 163–167, 2003.

[VBB03] A. Vincciarelli, S. Bengio, and H. Bunke. Offline recognition of large vocabulary cursive handwritten tex. In *In 7th international Conference on Document Analysis and Recognition*, volume 2, pages 1101–1105, 2003.

[Vin02] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1033–1446, 2002.

[VL00] A. Vinciarelli and J. Luettin. Off-line cursive script recognition based on continuous density hmm. In *7th International Workshop On Frontiers in Handwriting Recognition*, pages 1043–1050, Amsterdam, Holand, September 2000.

[VL03] A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 22(9):1043–1050, 2003.

[VTdlH+05] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R.C. Carrasco. Probabilistic finite-state machines. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 27(7):1013–1039, 2005.

[WB91] I.H. Witten and T.C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptative text compression. *IEEE Trans. on Information Theory*, 37(4):1085–1094, 1991.

[WCL94] S.S. Wang, P.C. Chen, and W.G. Lin. Invariant pattern recognition by moment fourier descriptor. *Pattern Recognition*, 27:1735–1742, 1994.

[WCW82] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document analysis system. *IBM J. Res. Devel.*, 26(6):647–656, 1982.

[WGJ+92] R.A. Wilkinson, J. Geist, S. Janet, P. Grother, C. Gurges, R. Creecy, B. Hammond, J. Hull, N. Larsen, and T. Vogl. The first census optical character recognition system conference. Technical report, Inst. of Standards and Technology, Gaithersburg, MD 20899, 1992.

[WMN99] F. Wessel, K. Macherey, and H.Ñey. A comparison of word graph and n-best list based confidence measures. In *European Conf. on Speech Technology (EuroSpeech)*, pages 315–318, 1999.

[WMR97] V. Wu, Manmatha, and E.M. Riseman. Finding text in images. In *Proceedings of the 2nd ACM International Conference on digital Libraries*, 1997.

[WMS98] F. Wessel, K. Macherey, and R. Schlüter. Using word probabilities as confidence measures. In *IEEE int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 225–228, 1998.

[WSMN01] F. Wessel, R. Schlüter, K. Macherey, and H.Ñey. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. on Speech and Audio Processing*, 9(3):288–298, 2001.

[WTS88] K. Wang, Y.Y. Tang, and C.y Suen. Multi-layer projections for the classification of similar chinese characters. *Pattern Recognition*, 1988.

[WY95] S.Y. Wang and T. Yagasaki. Block selection: A method for segmenting page image of various editing styles. In *3th International Conference on Document Analysis and Recognition*, pages 128–133, 1995.

[YJ96] Bin Yu and Amil K. Jain. A robust and fast skew detection algorithm for generic documents. *Pattern Recognition*, 29:1599–1629, 1996.

[YK03] Daekeun You and Gyeonghwan Kim. An efficient approach for slant correction of handwritten korean strings based on structural properties. *Pattern Recognition Letters*, 24:2093–2101, 2003.

[YS98] B. Yanikoglu and P. Sandon. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition Letters*, 31:1825–1833, 1998.

[ZB00] M. Zimmermann and H. Bunke. Automatic segmentation of the iam off-line database for handwritten english text. In *In Proc. of the 16th Int. Conf. on Pattern Recognition*, pages 35–39, 2000.

[ZB02] M. Zimmermann and H. Bunke. Hidden markov model lenght optimization for handwriting recognition systems. In *In Proc. of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 369–374, 2002.

[Zim03] Matthias Zimmermann. *Offline Handwriting Recognition and Grammar based Syntax Analysis*. PhD thesis, Philosophisch-naturwissenschaflichen Fakultï¿½t der Universitï¿½t Bern, Bern, November 2003.

[Zla94] A.A. Zlatopolsky. Automated document segmentation. *Pattern Recognition Letters*, 15(7):699–704, 1994.

[ZSW99] X. Zhu, Y. Shi, and S. Wang. A new algorithm of connected character image based on fourier transform. In *In Proc. of the 5th Int. Conf. on Document Analysis and Recognition*, pages 788–791, 1999.