

# Spontaneous Handwriting Recognition and Classification\*

Alejandro H. Toselli      Alfons Juan      Enrique Vidal

Instituto Tecnológico de Informática and  
Departamento de Sistemas Informáticos y Computación,  
ITI/DSIC, Universidad Politécnica de Valencia, 46071 Valencia, Spain.

E-mail: [ahector, ajuan, evidal]@iti.upv.es

## Abstract

*Finite-state models are used to implement a handwritten text recognition and classification system for a real application entailing casual, spontaneous writing with large vocabulary. Handwritten short paragraphs are to be classified into a small number of predefined classes. The paragraphs involve a wide variety of writing styles and contain many non-textual artifacts. HMMs and n-grams are used for text recognition and n-grams are also used for text classification. Experimental results are reported which, given the extreme difficulty of the task, are encouraging.*

## 1. Introduction

Cursive handwritten text recognition is currently becoming increasingly mature, thanks to the introduction of holistic approaches based on segmentation-free image recognition technologies. Using these techniques, very good results have been reported for applications entailing relatively clean and homogeneous handwriting and small vocabularies [4, 1, 3, 7, 9]. However, as the writing style becomes increasingly variable and spontaneous and/or the number of words to be recognized grows, performance of these systems tends to degrade dramatically.

Here we consider a handwritten text recognition and classification application entailing casual, spontaneous writing and a relatively large vocabulary. In this application, however, the extreme difficulty of text recognition is somehow compensated by the simplicity of the target result. The application consists of classifying (into a small number of predefined classes) casual handwritten answers extracted from survey forms made for a telecommunication company.<sup>1</sup>

As these answers were handwritten by a heterogeneous group of people, without any explicit or formal restriction

relative to vocabulary, the resulting application lexicon becomes quite large. On the other hand, since no guidelines are given as to the kind of pen or the writing style to be used, paragraphs become very variable and noisy.

Some of the difficulties involved in this application are worth mentioning. In some samples, the stroke thickness is non-uniform and the vertical slant also varies within a sample. Other samples present irregular and non-consistent spacing between words and characters. Also, there are samples written using different case and font types, variable sizes and even including foreign language phrases. On the other hand, noise and non-textual artifacts often appear in the paragraphs. Among these noisy elements we can find unknown words or words containing orthographic mistakes, as well as underlined and crossed-out words. Unusual abbreviations and symbols, arrows, etc. are also within this category. The combination of these writing-styles and noise may result in partly or entirely illegible samples. Examples of some of these difficulties are shown in Figure 1.

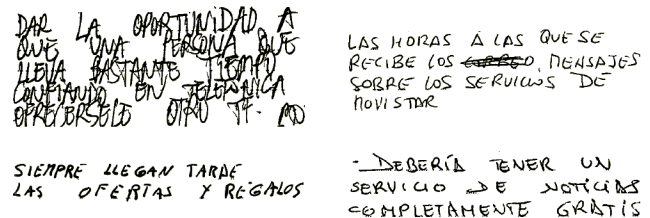


Figure 1. Examples of handwritten paragraphs of decreasing (form left to right, top to bottom) difficulty.

So far, human operators have been in charge of classifying these paragraphs. They do it through a fast reading which just aims to grasp the essential meanings of the answers. This implies that not all the words can or need to be perfectly recognized; they just retrieve enough information to get an adequate classification. The aim of our system is to help performing this classification as fast and accurately as possible, with a minimal human intervention.

In this paper we propose using the holistic, segmentation-free approach mentioned above to tackle

\* Work supported by the Valencian "Oficina de Ciencia i Tecnologia" under grant CTIDIA/ 2002/80 and Spanish "Ministerio de Ciencia y Tecnologia" under grant TIC2000-1 703-CO3-01.

<sup>1</sup> Data kindly provided by ODEC, S.A. ([www.odec.es](http://www.odec.es))

this difficult classification task. Finite-state models are homogeneously used to perform the task in two phases. First, using character HMMs and  $n$ -gram language models, recognition is performed on each handwritten sample; then, the recognized word sequence is classified into one of the given eight classes using a text classifier also based on  $n$ -grams. This work is based on [9], where both integrated and two-phase handwriting recognition and interpretation are discussed in detail. As previously mentioned, a simplification here is that interpretation consists just in classification into a small number of classes.

In the next section preprocessing and feature extraction are described. The adopted probabilistic framework is the topic of section 3. Section 4 considers the models which the system is based on. Experimental results are presented in section 5 and conclusions are drawn in the final section.

## 2. Preprocessing and Feature extraction

Only two processes take place in the preprocessing of each text image: noise reduction and line extraction. Because of the inherent difficulty of the task, line extraction is carried out so far in a semi-automatic way, based on a conventional line-extraction method [7]. Most of the paragraphs are processed automatically, but manual supervision is applied to difficult line-overlapping cases such as that shown in figure 1. By adequately pasting the extracted lines, a single-line (long) image is obtained.

As with any approach based on (one-dimensional) HMMs, feature extraction must transform the preprocessed image into a *sequence of (fixed-dimension) feature vectors*. To do this, the image is first divided into a grid of small square cells, sized a small fraction of the image height (such as 1/16, 1/20, 1/24 or 1/28). We call this fraction *vertical resolution*. Then each cell is characterized by the following features: *normalized grey level*, *horizontal grey-level derivative* and *vertical grey-level derivative*. To obtain smoothed values of these features, feature extraction is extended to a  $5 \times 5$ -cell window centered at the current cell and weighted by a Gaussian function. The derivatives are computed by least squares fitting a linear function.

Columns of cells are processed from left to right and a feature vector is built for each column by stacking the features computed in its constituent cells. An example of the resulting vector sequence is shown in figure 2. This process is similar to that followed in [1].

## 3. Probabilistic framework

Let  $\vec{x}$  be a sequence of feature vectors extracted from a handwritten line-image and let  $l$  identify the meaning of some text (just a classification label, in our case). The ultimate goal of our system is to find an optimal classification

for  $\vec{x}$ ; that is to search for an  $\hat{l}$ :

$$\hat{l} = \arg \max_l P(l | \vec{x}) \quad (1)$$

where  $P(l | \vec{x})$  is the posterior probability that  $l$  is the correct meaning (class) of  $\vec{x}$ .

Word recognition is not explicit in this formula, were recognition is seen as a hidden process. Nevertheless, classification can be viewed as a two-step process:

$$\mathbf{x} \rightarrow \mathbf{s} \rightarrow l$$

where  $s$  is a sequence of words. To uncover the underlying recognition process,  $P(l | \vec{x})$  can be seen as a marginal of the joint probability function  $P(s, l | \vec{x})$ . Using the Bayes rule and assuming that, in practice,  $p(\vec{x} | s, l)$  is independent of  $l$  we can write,

$$\hat{l} = \arg \max_l \sum_s P(s, l | \vec{x}) \quad (2)$$

$$= \arg \max_l \sum_s p(\vec{x} | s, l) P(s, l) \quad (3)$$

$$= \arg \max_l \sum_s p(\vec{x} | s) P(s, l) \quad (4)$$

Using  $P(s, l) = P(l | s)P(s)$  in eq. (4) and approximating the sum by the maximum, the optimization problem becomes:

$$(\hat{l}, \hat{s}) = \arg \max_{l, s} p(\vec{x} | s) P(l | s) P(s) \quad (5)$$

As shown in [9], if  $p(\vec{x} | s)$ ,  $P(l | s)$  and  $P(s)$  are modeled by finite state models, this problem can be directly solved efficiently. However, a two-step approximation is also possible and convenient to further reduce computation demands:

$$\hat{s} \approx \arg \max_c p(\vec{x} | s) P(s) \quad (6)$$

$$\hat{l} \approx \arg \max_l P(l | \hat{s}) = \arg \max_l P(\hat{s} | l) P(l) \quad (7)$$

where  $P(l)$  is the a priori probability of  $l$ .

Here we will follow this approach. For the first (recognition) step (eq. (6)), we adopt conventional HMMs to estimate  $p(\vec{x} | s)$  (as a sequence of character HMMs) and  $n$ -grams to estimate  $P(s)$ . For the second step (eq. (7)), a classifier is built by estimating  $P(\hat{s} | l)$  as an  $n$ -gram of words used in the class  $l$ .

Thanks to their *homogeneous* finite-state nature, in the recognition step both the HMMs and  $n$ -grams models can be easily integrated into a single *global* finite-state network [7] on which recognition can be efficiently performed. This is done by solving (6) using the well known Viterbi algorithm [5]. On the other hand, classification of a recognized word sequence is carried out by directly computing eq. (7) for each class  $l$  on the recognized text  $\hat{s}$  [2].

#### 4. Character, word and sentence modelling

Hidden Markov models have received significant attention in handwriting recognition during the last few years. As in speech recognizers for acoustic data [8], HMMs are used to estimate the probability for a sequence of feature vectors, seen as an “image realization” of a given text sentence. *Sentence* models are built by concatenation of *word* models which, in turn, are often obtained by concatenation of continuous left-to-right HMMs for individual *characters*.

Basically, each character HMM is a stochastic finite-state device that models the succession, along the horizontal axis, of (vertical) feature vectors which are extracted from instances of this character. Each HMM state generates feature vectors following an adequate parametric probabilistic law; typically, a *mixture of Gaussian densities*. The required number of densities in the mixture depends, along with many other factors, on the “vertical variability” typically associated with each state. On the other hand, the adequate number of states to model a certain character depends on the underlying horizontal variability. The possible or optional blank space that may appear between characters should be also modeled by each character HMM. In many cases the adequate number of states may be conditioned by the available amount of training data. Fig. 2 shows an example of character HMM.

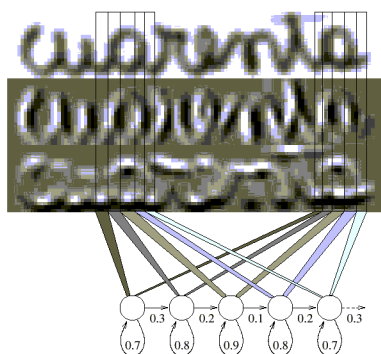


Figure 2. HMM modeling of instances of the character “a” within the Spanish word “cuarenta” (forty). The states are shared among all the instances of characters of the same class.

Once an HMM “topology” (number of states and structure) has been adopted, the model parameters can be easily trained from images of continuously handwritten text (*without any kind of segmentation*) accompanied by the transcription of these images into the corresponding sequence of characters. This training process is carried out using a well known instance of the EM algorithm called *forward-backward or Baum-Welch re-estimation* [5].

*Words* are obviously formed by concatenation of characters. In our finite-state modeling framework, for each word, a stochastic finite-state automaton is used to repre-

sent the possible concatenations of individual characters to compose this word. As previously discussed, the possible inter-character blank space is modeled by the character-level HMM. In contrast with continuous speech recognition, blank space often (but not always) appears between words. This automaton takes into account this possible blank space, as well as optional character capitalizations. An example of automaton for the Spanish word “mil” is shown in Fig. 3.

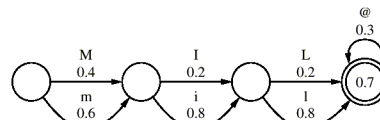


Figure 3. Automaton for the lexicon entry “mil” (One hundred). The symbol “@” represents a blank segment.

*Sentences* are formed by the concatenation of words. This concatenation is modeled by an  $n$ -gram model [5], which uses the previous  $n-1$  words to predict the next one; that is,

$$P(s) \approx \prod_{i=1}^N P(w_i | w_{i-n+1} \dots w_{i-1}) \quad (8)$$

where  $s = w_1 w_2 \dots w_N$  is a sequence of words.  $N$ -grams can be easily represented by finite-state deterministic automata.  $N$ -grams can be max-likelihood learned from a training (text) corpus, by simply counting relative frequencies of  $n$ -word sequences in the corpus [5].

As discussed in Section 3, all these finite-state (character, word and sentence) models can be easily *integrated* into a single *global* model on which equation 6 is easily solved; that is, given an input sequence of raw feature vectors, an output string of recognized words is obtained. Also, by implementing the classification phase in terms of  $n$ -gram models, classification is easily given by equation 7.

#### 5. Experiments

The image data-set extracted from survey forms consists of 913 binary images of handwritten paragraphs scanned at 300 dpi. Each of these images was preprocessed as discussed in Section 2. Following results reported in [9], a vertical resolution of 1/20 was adopted. Therefore, each paragraph image is represented as a sequence of  $(3 \times 20)$ -dimensional feature vectors. The resulting set of sequences was then partitioned into a training set of 676 samples and a test set including the 237 remaining samples.

In order to train the models described in section 4, two types of transcription of each training image were considered. One transcription set (TS-1), used to train the character HMMs and the recognition  $n$ -gram models, describes with detail and accuracy all the elements appearing in each handwritten text image, such as (lowercase

and uppercase) letters, symbols, abbreviations, spacing between words and characters, crossed-words, etc. The other transcription set (TS-2), used to train the eight classification  $n$ -gram models, was derived from TS-1 by converting lowercase characters to uppercase, eliminating the punctuation signs  $\{-\_/\#; :+* () | . , \sim ! ?\}$ , correcting orthographic mistakes and replacing abbreviations with their full-text equivalents. Details of these transcriptions sets and the partitions used in the experiments are shown in the table 1.

Number of:	Training	Test	Total	Lexicon
paragraphs	676	237	913	-
characters	64,666	21,533	86,199	80
words of TS-1	12,287	4,084	16,371	3308
words of TS-2	10,994	3,465	14,459	2118

**Table 1. Some Details about the image database as well as the training and test partitions. TS-1 and TS-2 refer to transcription set 1 and 2 respectively.**

All the 80 characters and symbols appearing in the image corpus were modeled using the same left-to-right HMM topology. After informally testing different values, they were configured with 6 states and 64 Gaussian (diagonal) densities per state. On the other hand, recognition 1-gram and 2-gram models using Witten-Bell back-off smoothing [6, 10] were trained from the TS-1 transcription set. In the same way, the eight classification 1-gram and 2-gram models were trained from the TS-2 transcription set.

Table 2 shows *recognition* Word Error Rates (WER) and *classification* errors for different combinations of recognition and classification  $n$ -gram models.

	Recognition WER(%)	Classification error(%)	
		1-gram(C)	2-gram(C)
1-gram(R)	56.1	52.3	57.8
2-gram(R)	54.3	52.7	56.1

**Table 2. Test-set recognition word error rate (WER) using unigram and bigram as language models and test-set classification error rate for different combinations of recognition (R) and classification (C) n-gram models.**

Although the recognition  $n$ -gram models were trained using TS-1, the test set recognition WER was determined by filtering the punctuation signs from all recognized word sequences and their respective test labels. The classification phase used this filtered recognition output.

2 LA SI HORAS A LAS FUESE RECIBE LOS VEO MENSAJES SOBRE LOS SERVICIOS DE MOVISTAR  
 3 SIEMPRE LLEGAN TARDE LAS OFERTAS TRES AOS  
 4 DEBERA TENER UN SERVICIO DE NOTICIAS COMPLETAMENTE GRATIS

**Table 3. Recognized paragraphs from Figure 1.**

Table 3 shows examples of recognition results corresponding to paragraph images shown in Figure 1. There was no output for the first image, the second and third ones produced three and two word errors, respectively and the last one was perfectly recognized.

## 6. Conclusions

A classifier of spontaneous handwritten paragraphs based on Hidden Markov Models and  $n$ -grams is presented. Because of the large vocabulary of the application and the limited number of training samples available, recognition and classification  $n$ -gram models are clearly undertrained. Despite this sparse data condition, and given the extreme difficulty of the task (which in many ways resembles spontaneous speech recognition), the preliminary results achieved in this work are encouraging.

## References

- [1] I. Bazzi, R. Schwartz, and J. Makhoul. An Omnifont Open-Vocabulary OCR System for English and Arabic. *IEEE Trans. on PAMI*, 21(6):495–504, 1999.
- [2] W. B. Cavnar and J. M. Trenkle.  $n$ -gram-based text categorization. In *Proc. of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94)*, pages 161–175, Las Vegas, Nevada, U.S.A., Apr. 1994.
- [3] 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 the S+SSPR 2000*, pages 143–153, Alicante (Spain), 2000.
- [4] D. Guillevic and C. Y. Suen. Recognition of legal amounts on bank cheques. *Pattern Analysis and Applications*, 1(1):28–41, 1998.
- [5] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.
- [6] S. 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*, ASSP-35:400–401, Mar. 1987.
- [7] U.-V. Marti and H. Bunke. Using a Statistical Language Model to improve the performance of an HMM-Based Cursive Handwriting Recognition System. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.
- [8] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1993.
- [9] A. H. Toselli, A. Juan, D. Keysers, J. Gonzalez, I. Salvador, H. Ney, E. Vidal, and F. Casacuberta. Integrated Handwriting Recognition and Interpretation using Finite-State Models. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2003. Accepted.
- [10] I. H. Witten and T. C. Bell. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Trans. on Information Theory*, 17(4), July 1991.