

Offline Recognition of Syntax-Constrained Cursive Handwritten Text

J. González, I. Salvador, A.H. Toselli, A. Juan, E. Vidal, and F. Casacuberta

Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia,
Camino de Vera s/n, 46071 Valencia, Spain
Phone: (34) 96 387 7240
Fax: (34) 96 387 7239

{jgonza, isalva, ahector, ajuan, e Vidal, fcn}@iti.upv.es

Abstract. The problem of *continuous handwritten text* (CHT) recognition using standard *continuous speech recognition* technology is considered. Main advantages of this approach are a) system development is completely based on *well understood training techniques* and b) *no segmentation* of sentence or line images into characters or words is required, neither in the training nor in the recognition phases. Many recent papers address this problem in a similar way. Our work aims at contributing to this trend in two main aspects: i) We focus on the recognition of *individual, isolated characters* using the very same technology as for CHT recognition in order to tune essential representation parameters. The results are themselves interesting since they are comparable with state-of-the-art results on the same standard OCR database. And ii) *all* the work (except for the image processing and feature extraction steps) is strictly based on a well known and widely available *standard toolkit* for *continuous speech recognition*.

Keywords: Off-Line Continuous Handwriting Text Recognition, Feature Extraction, Language Modelling, Hidden Markov Models, Bank Check Legal Amount Recognition

1 Introduction

The recognition of off-line, continuously handwritten text is proving to be a quite challenging pattern recognition task. Although text is basically composed of characters, most traditional approaches to *optical character recognition* (OCR) generally fail in this task because of the extreme difficulty of segmenting continuously written text into characters. In fact, not even the segmentation into words can be reliably accomplished using standard techniques in most cases. Nevertheless, humans do accurately perform both segmentation and recognition in a seemingly effortless manner. Accurateness is achieved by “delaying” recognition until the highest perception level: only after having understood a written (part of a) message are humans capable to “recognize” the constituent words,

the corresponding characters and the underlying segmentations. Clearly, this streaking human ability comes from a tight cooperation of *morphologic*, *lexical*, *syntactic* and *semantic-pragmatic* knowledge to accomplish the task.

Just this very same situation appears in the field of *continuous speech recognition* (CSR) [1]. In this field, successful techniques already exist which are actually based on approaching the abovementioned tight cooperation of knowledge sources. After many decades of research in this field, commonly accepted adequate solutions come from three basic principles: i) adopt *simple*, *homogeneous* and easily understandable models for all the knowledge sources, ii) formulate the recognition process as an *optimal search* through an adequate structure based on these models, and iii) use adequate techniques to *learn* the different models from training data of each considered task. All these principles are properly fulfilled by the use of *finite-state* (FS) modeling techniques such as *hidden markov models* (HMM) and *stochastic FS* (SFS) *grammars* or *automata* [2].

In this paper, we address the problem of *continuous handwritten text* (CHT) recognition using *standard* CSR technology. Many recent papers address this problem in a similar way (see, among other [3,4,5,6]). Our work aims at contributing to this trend in two main aspects: i) As an important part of the work, we focus on the recognition of *individual, isolated characters* using the very same FS technology as for CHT recognition. This study parallels similar work on phonetic decoding that has proved quite helpful to optimize CSR systems. The results are themselves interesting since they are comparable with state-of-the-art results on the same standard OCR database. And ii) *all* the work (except for the image processing and feature extraction steps) is strictly based on the well known and widely available *standard HTK toolkit* for CSR [7]. As an application example, in this work we focus on the recognition of legal amounts in bank checks.

2 Feature Extraction

Following current trends in HMM-based off-line handwriting text recognition [6], the image of a text sentence or line is represented as a *sequence of feature vectors*. The height of the image is first normalized to a constant value so as to minimize the dependence on writing style size. Then, the image is divided into a grid of squared cells whose size is a small fraction of the image height (tested values are 1/16, 1/20 and 1/24). Each cell is characterized by the following simple and script-independent features: *normalized grey level*, *horizontal grey-level derivative* and *vertical grey-level derivative*.

To obtain smoothed values of these features, feature extraction is not restricted to the cell under analysis but extended to a 5×5 window centered at the current cell. To compute the normalized grey level, the analysis window is smoothed by convolution with a 2-d Gaussian filter. On the other hand, the horizontal grey-level derivative is computed as the slope of the line which best fits the horizontal function of column-averaged grey levels. The fitting criterion is the sum of squared errors weighted in accordance with a 1-d Gaussian filter

which enhances the role of central pixels in each analysis window. The vertical grey-level derivative is computed in a similar way.

Columns of cells or *frames* are processed from left to right and a feature vector is constructed for each frame by concatenating the features computed in its constituent cells (see fig. 1).

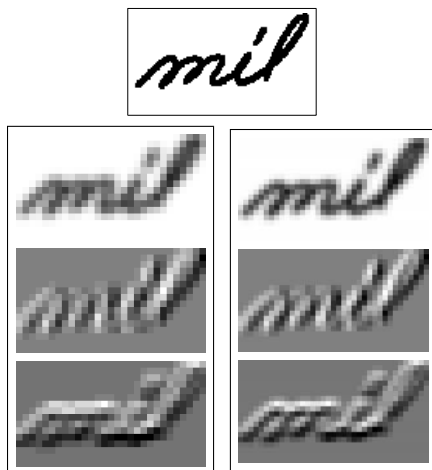


Fig. 1. Feature extraction for the Spanish Sentence “*mil*” (one thousand). Top: Original image; bottom: two representations into sequences of k -dim vectors; left $k = 16 \times 3$, right $k = 20 \times 3$. Each (column) vector is divided into three blocks (from top to bottom): normalized grey levels, horizontal derivatives and vertical derivatives.

3 Character, Word, and Sentence Modelling

Individual *characters* are modelled by *continuous density left-to-right hidden Markov models* (HMM), similar to those used in CSR [1]. Fig. 2 shows an example of the structure of one of these models. Basically, each character HMM is a SFS device that has to model the succession, along the horizontal axis, of (vertical) feature vectors which are extracted from instances of this character. It is assumed that 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. This number needs to be empirically tuned in each task. On the other hand, the number of states that is adequate to model a certain character or character set depends on the underlying “horizontal variability”. For instance, to ideally model a capital “E” character, only two states might be enough (one to model the vertical bar and the other for the three horizontal strokes), while three

states may be more adequate to model a capital “H” (one for the left vertical bar, another for the central horizontal stroke and the last one for the right vertical bar). Note that the possible or optional blank space that may appear between characters should be also modelled by each character HMM. The most appropriate number states for a given task also depends of the amount of training data which is available to train model parameters. So, the exact number of states to be adopted needs some empirical tuning in each practical situation. Once a HMM “*topology*” (number of states and structure) has been adopted, the model parameters can be easily trained from continuously written text (*without any kind of segmentation*) accompanied by the transcription of this text into the corresponding sequence of characters (c.f. Sect. 5.2, 5.3). This training process is carried out using a well known instance of the EM algorithm called *backward-forward or Baum-Welch re-estimation* [1]. Obviously, the very same technique can also be used if isolated versions of the individual characters are available (c.f. Sect. 5.1), as in standard OCR.

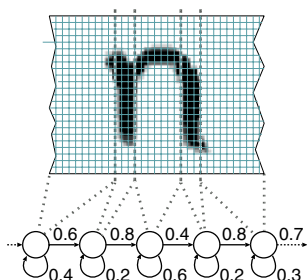


Fig. 2. Structure of a Character Left-to-Right Hidden Markov Model aimed at modelling instances of the character “n”.

Words are obviously formed by concatenation of characters. In our FS modeling framework, for each word, a SFS automaton is used to represent the possible concatenations of individual characters to compose this word. This automaton also takes into account optional capitalizations, as well as the blank space usually left at the end of each word (as previously discussed, the possible inter-character blank space is modeled at the character level HMM). An example of automaton for the Spanish word “*mil*” is shown in Fig. 3.

Sentences are formed by the concatenation of words. In contrast with CSR, blank space often (*but not always*) appears between words. As previously discussed, this optional blank space is modeled at the lexical level. The concatenation of words is modeled by a (FS) *language model*. In our bank check example application, it consist in a FS grammar which recognizes all the text written Spanish numbers from 0 to $10^{12} - 1$. The terminal symbols (or *lexicon*) of this grammar are Spanish words used to write numbers, such as “*uno*”, “*dos*”, “*diez*”, “*sesenta*”, “*cien*”, “*mil*”, “*millón*”, etc. (one, two, ten, sixty, hundred, thousand, million, etc). Moreover, this model is built as a *sequential FS transducer*

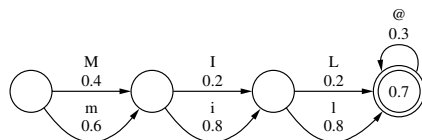


Fig. 3. Automaton for the lexicon entry “*mil*”. The symbol “@” is for a *blank* segment.

which also provides an output for each input sequence of words. The output is an *arithmetic expression* whose value is that of the number given through the input text; for example, from the Spanish text “*doscientos sesenta y dos mil veinte*” (two hundred sixty two thousand and twenty) the obtained output is: “ $+(200 + 60 + 2) * 1000 + 20$ ”. From this expression the target (decimal) number (262,020) can be easily obtained. A small fragment of this transducer is shown in Fig. 4.

The aim of this setting is similar to that in [5]. However the approach followed here is strictly based on FS technology and is therefore much simpler. In fact, in our system the required decimal digit string is just obtained by directly piping the output of the CHT recognizer to the standard Unix tool “bc”. Furthermore, (in most languages) this last evaluation step can be avoided all together by the use of a slightly more powerful kind of FS devices known as “*subsequential*” *transducer* [8]. These FS devices, which are automatically learnable from training data [8], allow *direct* translation of text-represented numbers into decimal form. In this way, the use of *syntax-directed transducers* as in [5] (which are *not* FS and would therefore break our *homogeneity* assumption) is no longer needed.

Some features of our *numbers transducer* are: 51 input words, 187 output tokens, 32 states, 660 transitions; (*test-set*) Perplexity: 6.2.

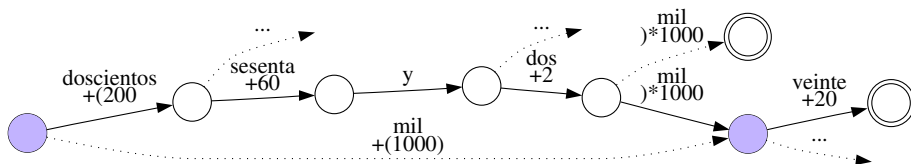


Fig. 4. A piece of the *numbers transducer*. Solid-line edges correspond to a path that accepts “*doscientos sesenta y dos mil veinte*” (two hundred sixty two thousand and twenty), yielding “ $+(200+60+2)*1000+20$ ”.

4 Knowledge Integration: Recognition as a Best Hypothesis Search

Once all the *character*, *word* and *language* models are available, recognition of new test sentences can be performed. Thanks to the *homogeneous* FS nature of all these models, they can be easily *integrated* into a single *global* (huge) FS model that accepts sequences of raw feature vectors and outputs strings of recognized words (and, in our application, also the corresponding arithmetic expressions). Fig. 5 illustrates this integration.

Given an input sequence of feature vectors, the best output hypothesis is one which corresponds to a series of states of the integrated model that, with highest probability, produces the input feature-vector sequence. This global search process is very efficiently carried out by the well known (*beam-search*-accelerated) Viterbi algorithm [1]. This technique allows integration to be performed “on the fly” during the decoding process. In this way, only the memory strictly required for the search is actually allocated.

5 Experiments

Experiments have been carried out to tune certain constants at each knowledge level, to train HMM parameters and to test the recognition performance of the resulting systems.

5.1 Isolated Character Recognition: Optimizing Feature Extraction Parameters

Although accurate recognition requires knowledge at higher perception levels, *isolated* character recognition serves as a good basis to make adequate decisions on feature extraction parameters. To this end, rather simple features were empirically compared in order to assess their discriminating power in the classification of the 18 lowercase letters appearing in our bank check application (“a, c, d, e, h, i, l, m, n, o, q, r, s, t, u, v, y, z”).

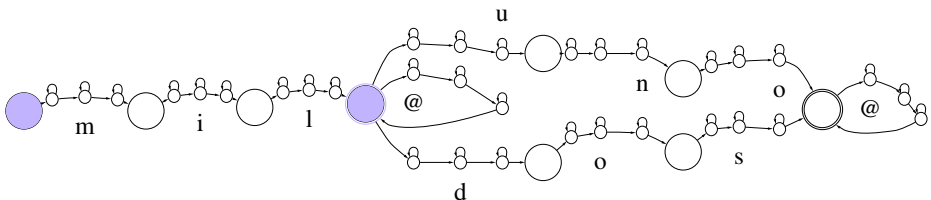


Fig. 5. A small piece of an integrated FS model, using three-state character HMMs. The part shown stands for the sentences “mil”, “mil uno” and “mil dos” (1,000; 1,001; 1,002). For the sake of clarity, only un-capitalized word models are shown and output arithmetic-expression tokens are omitted.

Letter samples were extracted from the widely used NIST Special Database 3 [9]. This database, published on CDROM [10], includes 45,313 binary images of (segmented) lowercase letters extracted from forms written by 2,100 writers. For our purposes, a moderate-size training set was built out of samples of our 18 lowers from the first 200 writers (overall 3,034 samples, one sample of each lower per writer, when available). Similarly, a test set of 750 samples was extracted from 50 independent writers (writers #1,051 through #1,100). Following the procedure described in section 2, six different image representations were considered by combining two sets of features (grey-level alone or grey-level plus derivatives) and three vertical resolutions (1/16, 1/20 or 1/24).

Left-to-right four-state continuous-density HMMs were used to model each character. Each state had assigned a mixture of k Gaussian densities with diagonal covariance matrices. Values of k in $\{1, 2, 4, 8, 16, 32\}$ were tried in the experiments. HMMs were trained and tested using the HTK toolkit [7]. More precisely, three cycles of Baum-Welch re-estimation were run to train model parameters, while the Viterbi recogniser was used to classify each test sample in accordance with the most likely HMM. Results are shown in Table 1.

Table 1. Classification error rate (in %) for isolated character recognition, using four-state continuous-density HMMs, different feature sets (with/out derivatives) and varying number of Gaussian densities per model-state.

Set of features	Vertical resolution	Number of Gaussian densities per state					
		1	2	4	8	16	32
Grey Level	1/16	33.8	28.2	24.0	23.4	21.0	20.9
	1/20	31.3	26.8	20.7	19.0	17.9	16.6
	1/24	31.0	26.8	23.1	19.5	17.2	14.1
Grey Level & Derivatives	1/16	21.9	16.7	14.6	12.3	10.9	10.5
	1/20	20.1	17.9	13.4	11.1	9.5	11.0
	1/24	21.2	18.4	14.3	11.8	9.2	10.1

From these results, it is clear that using grey-level derivatives significantly improves recognition accuracy. Vertical resolution, however, does not seem to be a key factor: a vertical resolution of 1/20 might be good enough for the experiments with continuous text. On the other hand, it is worth noting that our best error rates (9.2% and 9.5%) are similar to the best figures reported (at zero-rejection rate) after the First Census OCR Conference (11% for the best system and 8.6% for two-pass human classification) [11, p. 26]. Although we do not face classification of the whole set of lowers, these results encourage us to continue exploring the application of FS technology to OCR.

5.2 Recognition of Sentences Composed of Artificially Concatenated Characters: Assessing the Power of Model Integration

The data for the second series of experiments consisted of 500 images of random sentences composed by the concatenation of the appropriate randomly selected

setenta y cuatro millones
 treinta y seis mil ochenta
 mil setecientos millones veintidos mil veintisiete

Fig. 6. Examples of sentences produced by concatenating randomly selected handwritten characters from the NIST database: 74,000,000; 36,080; 1,700,022,027.

handwritten isolated characters. Overall, these images contained 9,504 characters, which were drawn from the same corpus as that of the previous experiment. The sentences corresponded to simulated legal amount numbers in bank checks (see examples in Fig. 6). From this data, 313 sentences (609 words, 5,900 characters) were devoted to train the character models and 187 sentences (975 words, 3,604 characters) to test the performance of the recognition system. No data from “training writers” were used in the composition of test sentences.

The main difference between this setting and that of the previous experiment is that now training (and testing) is carried out using long images of continuous text, without any kind of segmentation or information about the actual position of the characters in each sentence. The aim of this experiment was to assess the power of integrating morphologic, lexical and language models to improve recognition performance.

Automatically determining a different topology and/or number of states which is best suited to model each particular character proves to be a non-trivial problem. Therefore, in this work, identical topology was adopted for all the characters. Left-to-right continuous-density HMMs of N states and k Gaussian densities per state ($N \in \{4, 6, 8\}$, $k \in \{1, 2, 4, 8, 16, 32, 64\}$) were used for character modeling plus a special model for the blank character (“@” in Fig. 3 and 5). In this case, the training procedure was the usual one for acoustic modeling of phone units in continuous speech: character-level HMMs were trained through four iterations of the Baum-Welch algorithm. This process was initialized by a linear segmentation of each training image into a number of equal-length segments according to the number of characters in the orthographic transcription of the sentence. As in the previous experiments, these models were trained using the HTK toolkit. For each test input sentence, the Viterbi decoding algorithm was performed on the FS network which integrates character, lexicon and language models.

Test-set recognition *word error rates* (WER) are presented in Table 2. Best results were achieved using 4-state character HMMs with 8 and 16 Gaussian densities per state. In this case, a WER of 3.0% was obtained. If compared with the isolated character error rate (5.1), these results clearly show the power of model integration. The corresponding *digit error rate*, obtained by evaluating the arithmetic expression obtained as the translation of each recognized sentence, was 2.8%, with 1.8% substitution errors, 0.9% digit deletions and 0.1% insertions. From the 187 test sentences, 14 (7.5%) yielded decimal numbers with one (or more) digit(s) in error. Digit error rates are the relevant results if the output

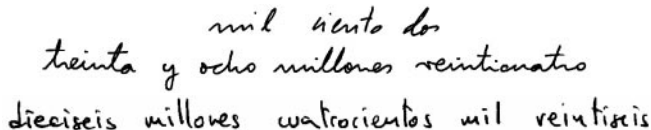
of legal amount recognition is to be validated with the help of the OCR results obtained from the corresponding courtesy amount in the same bank check.

Table 2. Test set recognition word error rates (in %) for artificially concatenated characters. Results for different numbers of states per model and Gaussian densities per state are reported.

States	Gaussian densities per State						
	1	2	4	8	16	32	64
4	8.5	5.3	3.3	3.0	3.0	3.3	7.6
6	6.1	3.8	3.1	3.5	3.3	3.9	10.2
8	14.4	10.6	9.1	7.8	7.0	10.1	17.2

5.3 Recognition of Real Continuous Text Sentences

The corpus for this experiment was composed by 485 real images of handwritten Spanish legal amounts (2,127 words, 16,039 characters), handwritten by 29 different writers (see Fig. 7). 298 randomly selected sentences from 18 writers were used for training and 187 from the rest of the writers were left for testing.



mil ciento do
treinta y ocho millones veintianatro
dieciseis millones cuatrocientos mil veintiseis

Fig. 7. Examples of real continuous text sentences: 1,102; 38,000,024; 16,400,026.

The training and testing procedures were the same as those described in section 5.2. Test-set recognition Word Error Rates (WER) are presented in Table 3. Graphic results for the best number of states (5) and the best number of densities per state (16) are also shown in Fig. 8. For the best setting, a WER of 18.0% and a DER of 13.2% were obtained. These good results clearly assess the adequateness of the proposed technology for continuous handwritten text recognition.

6 Conclusion

The problem of CHT recognition can be adequately addressed using standard CSR technology on images that are represented by sequences of fairly simple feature vectors.

An image is divided into a sequence of vertical sets of cells. Each cell is represented by the normalized grey level in its vicinity and the corresponding

Table 3. Test-set recognition word error rates (in %) for continuous handwritten sentences of legal amount numbers in bank checks. Results for different number of states per model and Gaussian densities per state are reported.

Number of gaussians per state	Number of states per model				
	3	4	5	6	7
1	77.4	61.1	55.5	52.6	52.1
2	69.2	48.6	39.8	35.7	36.6
4	55.3	37.2	27.0	25.6	28.5
8	44.3	30.5	21.3	19.3	25.9
16	38.0	26.0	18.0	18.6	24.5
32	33.9	24.2	18.4	20.1	25.3
64	36.3	26.5	23.1	25.5	33.9

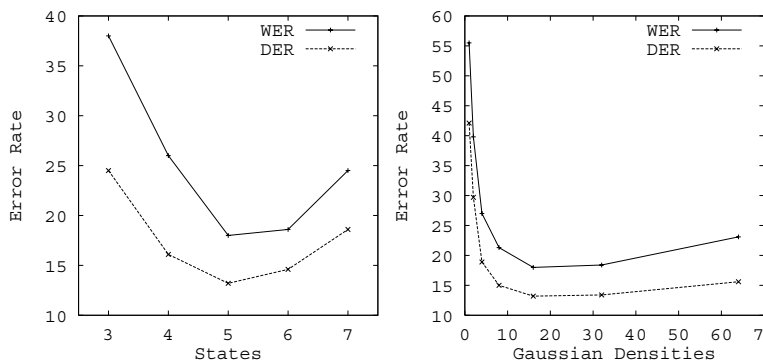


Fig. 8. Test-set recognition word (and digit) error rates (in %) for the continuous handwritten sentences of legal amount numbers in bank checks. Results for different number of states per model and 16 Gaussian densities per state (left) and results for different number of Gaussian densities per state and 5 states per model (right) are shown.

horizontal and vertical grey-level derivatives. Using these vector sequences, *character hidden markov models* can be trained by the well known Baum-Welch re-estimation algorithm. New handwritten text can then be recognized through the standard Viterbi decoding algorithm on a (virtually) integrated FS *network* composed by character, lexicon and syntactic FS models.

This methodology has been tested on the recognition of isolated characters (OCR) with quite competitive results. Also, experiments on the recognition of legal amounts in bank checks have been performed, with very promising results. These results, however, should only be considered preliminary, since they have been obtained without the help of simple normalisation preprocessing procedures that are quite standard in this task. Work is currently under way to include two

of these procedures, namely, *slant normalisation* and *dynamic baseline detection*, with well known potential for significant performance improvements.

References

- [1] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.
- [2] E. Vidal. Language learning, understanding and translation. In R. de Mori H. Niemann and G. Hanrieder, editors, *CRIM/FORWISS Workshop on Progress and Prospects of Speech Research and Technology*, Proc. in Art. Intell., pages 131–140. Infix, 1994.
- [3] D. Guillevic and C. Y. Suen. Recognition of legal amounts on bank cheques. *Pattern Analysis and Applications*, 1(1):28–41, 1998.
- [4] D. Guillevic and C. Y. Suen. HMM-KNN Word Recognition Engine for Bank Cheque Processing. In *ICPR 98*, volume 2, pages 1526–1529, Brisbane (Australia), August 1998.
- [5] G. Kaufmann and H. Bunke. Amount Translation and Error Localization in Check Processing Using Syntax-Directed Translation. In *ICPR 98*, volume 2, pages 1530–1534, Brisbane (Australia), August 1998.
- [6] I. Bazzi, R. Schwartz, and J. Makhoul. An Omnifont Open-Vocabulary OCR System for English and Arabic. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(6):495–504, June 1999.
- [7] S.J. Young, P. C. Woodland, and W.J. Byrne. HTK: Hidden Markov Model Toolkit V1.5. Technical report, Cambridge University Engineering Department Speech Group and Entropic Research Laboratories Inc., 1993.
- [8] J. Oncina, P. García, and E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-15(5):448–458, May 1993.
- [9] M. D. Garris and R. A. Wilkinson. Handwritten segmented characters database. Technical Report Special Database 3, NIST, February 1992.
- [10] R.A. Wilkinson and M.D. Garris. Handprinted Character Database. Technical report, NIST, April 1990.
- [11] J. Geist et al. The Second Census Optical Character Recognition Systems Conference. Technical Report NISTTR 5452, NIST, May 1994.