Offline Grammar-Based Recognition of Handwritten Sentences

Matthias Zimmermann, Jean-Cédric Chappelier, and Horst Bunke, *Member*, *IEEE*

Abstract—This paper proposes a sequential coupling of a Hidden Markov Model (HMM) recognizer for offline handwritten English sentences with a probabilistic bottom-up chart parser using Stochastic Context-Free Grammars (SCFG) extracted from a text corpus. Based on extensive experiments, we conclude that syntax analysis helps to improve recognition rates significantly.

Index Terms—Optical character recognition, handwriting analysis, natural language parsing and understanding.

1 Introduction

In the field of offline handwriting recognition, we observe a tendency to address problems of increasing complexity. High recognition rates have been published for the recognition of isolated digits [25] or characters [29]. The recognition performance achieved for isolated words [20] is already significantly lower. If the task complexity increases further, as in the case of the recognition of handwritten addresses [21] or bank checks [9], task specific knowledge like the relation between zip code and city name, or between courtesy amount and legal amount, becomes essential. For general text recognition, task specific information can be found in the linguistic domain. The successful application of word n-gram language models supporting the recognition of handwritten text lines has been reported in [22], [30]. However, the effectiveness of n-gram language models is limited to short distance relationships between words.

In this paper, we try to overcome these shortcomings with a sequential coupling of a recognition system for handwritten English sentences (see Fig. 1) and a syntax analysis module based on Stochastic Context-Free Grammars (SCFGs). The goal of our approach is to improve the performance of the recognition system and to create additional linguistic information in the form of grammatical word tags (e.g., noun, pronoun, verb form) as well as parse trees. Parse trees represent the hierarchical structure of the grammatical constituents of a sentence (e.g., noun phrases, verb phrases, adjective phrase). Such extra information can be valuable in various contexts, for example, semantic information retrieval or text understanding. To the knowledge of the authors, it is the first time that linguistic information in form of an SCFG has been applied in the field of handwriting recognition. An early version of the paper has been published in [36]. The current paper provides more details and results are based on much larger experiments.

The rest of the paper is organized as follows: Section 2 reviews related work. The methodology is described in Section 3, while experiments and results are reported in Section 4. Conclusions are drawn in the last section of this paper.

- *M. Zimmermann is with the International Computer Science Institute (ICSI), 1947 Center Street, Suite 600, Berkeley, CA 94704-1198. E-mail: zimmerma@icsi.berkeley.edu.*
- *J.-C. Chappelier is with the Swiss Federal Institute of Technology, EPFL/IC/IIF/LIA, Batiment IN, Station 14, CH-1015 Lausanne, Switzerland. E-mail: jean-cedric.chappelier@epfl.ch.*
- *H. Bunke is with the Institute of Computer Science and Applied Mathematics (IAM), University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland. E-mail: bunke@iam.unibe.ch.*

Manuscript received 7 June 2004; revised 9 Sept. 2005; accepted 6 Oct. 2005; published online 13 Mar. 2006.
Recommended for acceptance by T. Tan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0287-0604.

0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

2 Related Work

In the past, a number of different approaches involving syntax analysis to improving recognition rates were proposed in the domains of speech and Optical Character Recognition (OCR). In the case of OCR, only a few publications investigate the use of syntax analysis. The use of linear grammars is described in [12], [28]. Sets of valid syntactic patterns are utilized in [4] and a word lattice rescoring mechanism is proposed in [17]. In [11], a Context-Free Grammar (CFG) is used to improve word recognition rates. The use of syntactical knowledge is more widespread in the domain of speech recognition. In earlier works [18], [19], CFG are used. More recently, Stochastic Context-Free Grammars (SCFG) are becoming more common [2], [6], [15], [27].

The highest performance improvements found in the literature are reported in [11], [12], [15]. However, these works make use of relatively small grammars explicitly written for specific tasks which do not have to deal with the full amount of ambiguity present in natural language.

References [3], [6], [27] are closely related to the topic and the experimental setup of this paper. These works combine of a word trigram language model and a broad coverage SCFG. Results are based on 213 sentences from the DARPA '93 HUB1 test setup and performance improvements are reported against the baseline word trigram language model. In [3], the word error rate is reduced from 13.7 percent to 13.0 percent, [6] measures a reduction of the word error rate from 16.6 percent to 16.0 percent, and [27] reports a word error rate reduction from 16.5 percent to 15.1 percent.

3 Methodology

We first explain the recognition of handwritten text and the extraction of the grammar. Then, parsing of English sentences is introduced before we describe the proposed combination scheme for the recognizer and the parser.

3.1 Offline Recognition of Handwritten Sentences

The goal of handwritten text recognition is to find the most likely sentence $\widehat{W} = (w_1 \dots w_n)$ for a given feature vector sequence $X = (X_1 \dots X_m)$, i.e., $\widehat{W} = argmax_W P(W|X)$. The application of the Bayes' rule leads to a decomposition of $P(W|X)$ into the *optical model* $P(X|W)$ and a *statistical language model* $P(W)$. The problem can then be reformulated as one of finding the word sequence $\widehat{W}$ that maximizes a sentence score $\phi(W)$:

$$\widehat{W} = \underset{W}{argmax}\, \phi(W), \tag{1}$$

$$\phi(W) = \log P(X|W) + \log P(W). \tag{2}$$

In our case, $P(X|W)$ is estimated by a recognizer based on the Hidden Markov Model (HMM) technique [26] which is supported by a word bigram language model to approximate $P(W)$. The word bigram language model is integrated in the recognition process using the two parameters α and β commonly applied in the domain of speech recognition (e.g., [24]). This leads to a modified sentence score $\phi(W)$[1]

$$\phi(W) = \log P_{HMM}(X|W) + \alpha \log P_{BG}(W) + n\beta. \tag{3}$$

The two parameters α and β help to overcome deficiencies of the likelihood values $P_{HMM}(X|W)$ from the HMMs and the probabilities $P_{BG}(W)$ provided by the word bigram language model. In (3), parameter α weights the influence of the language model against the optical model and parameter β helps to control word insertion and deletion rates. For large values of β, the recognizer will favor candidate sentences containing many short words. Small or negative values of β will have the opposite effect and lead to sentences containing fewer and longer words. Optimal values of parameters α and β are determined by experiment using a validation set.

1. The number of words in candidate sentence W is represented by n.

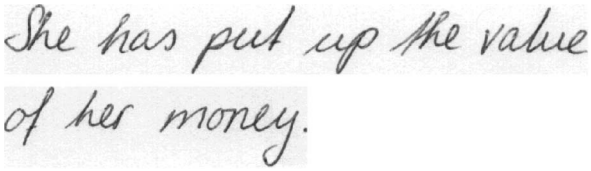

Fig. 1. An automatically extracted sentence from the IAM database.

Rank	$\phi(W)$	Candidate sentence
1	23,924	She has put up the value other money .
2	23,922	She has put up the value of her money .
3	23,890	She had put up the value other money .
4	23,888	She had put up the value of her money .
5	23,854	She has put up the value at her money .

Fig. 2. An example of an n-best list for the first five candidate sentences with corresponding recognition scores $\phi(W)$ for the sentence shown in Fig. 1.

The handwritten sentence recognition system used in this paper is an enhanced and optimized version of [22]. Recognition is performed in three major steps: text line normalization, extraction of feature vector sequences, and decoding. In the feature extraction step, a sliding window is used to produce a sequence of feature vectors (observations) from the normalized text line images. For each image column, a feature vector containing nine geometrical features is extracted, e.g., the number of foreground pixels in the window, moments of the foreground pixels, etc. To model the 85 characters considered in our application, continuous density HMMs with a linear topology are used. The character set includes lower and uppercase letters, digits, interpunctuation, and some special characters found in the texts to be recognized. Compared to the original system [22] the number of model states is optimized per character [34] and Gaussian mixtures are used for the emission probabilities instead of single Gaussians. The main recognition step consists in Viterbi decoding [31] which is supported by a word bigram language model. For language model smoothing, we use the Good-Turing technique [8] together with Katz-backoff to lower order models [16]. In contrast to other works in the domain of handwriting recognition, the integration of the word bigram language model is optimized as described in [35]. The result of the recognition process described above consists in a list of the n-best candidate sentences for a given input sentence image (see Fig. 2).

3.2 Grammar Extraction

An SCFG is a five-tuple $(N, T, P, S, p(.))$, where N represent the set of nonterminal symbols and T the set of terminal symbols, such that $N \cap T = \emptyset$. Nonterminal symbols typically represent syntactical categories, i.e., word tags and sentence constituents. Terminal symbols correspond to the words in the lexicon and $S \in N$ defines the start symbol. All productions $\in P$ are written as $A \rightarrow \alpha$, where $A \in N$ and $\alpha \in (N \cup T)^+$. Productions of the form $A \rightarrow w$ with $w \in T$ are called *lexical productions*. The probability function $p(.)$ maps productions to the interval $(0, 1]$, such that $\sum_\alpha p(A \rightarrow \alpha) = 1$ for each $A \in N$.

In practice, SCFGs can be extracted from special text corpora called treebanks which contain parse trees in the form of bracketed sentences. Based on the bracketed notation, it is straightforward to extract the corresponding productions using a simple push down automaton. Production probabilities are then estimated from the relative frequencies according to (4) below, where $\#(A \rightarrow \alpha)$ represents the number of times that production $A \rightarrow \alpha$ is observed in the treebank.

$$p(A \rightarrow \alpha) = \frac{\#(A \rightarrow \alpha)}{\sum_\beta \#(A \rightarrow \beta)}. \quad (4)$$

In order to estimate the probabilities of the lexical productions tagged, corpora can be used. From the tagged words (A, w), the productions $A \rightarrow w$ are directly derived, where A represents the grammatical word tag and w the word itself. The corresponding probabilities are estimated using $p(A \rightarrow w) = \#(A, w)/\#(A)$, where the number of occurrences of the tagged word (A, α) is measured by $\#(A, w)$, and $\#(A)$ represents the number of times the word tag A has been observed in the tagged Lancaster-Oslo/Bergen corpus [13].

3.3 Parsing English Sentences

For the syntax analysis of the n-best candidate, sentences provided by the handwriting recognition module, a bottom-up chart parsing algorithm for SCFGs is used. This algorithm is detailed in [1] and can be seen as an extension of the algorithms presented in [5], [10], [32]. It is able to compute the probability of the input sequence W as well as probability $P_{SCFG}(W)$ of its most probable parse, and to find, with their probabilities, all parses of all subsequences of the input sequence. It also deals, in a probabilistic way, with multiple interpretations of sentences containing compound words.

Like most parsing algorithms, our parser is not only a recognizer determining whether an input sequence is syntactically correct or not, but also an analyzer, producing a compact representation of all parses for all the subsequences of the input. It is particularly easy to extract the most-probable parse tree from the chart associated with the input sequence. The computation of the most probable parse and its probability $P_{SCFG}(W)$ in the bottom-up phase of the algorithm is very useful for our application since it supports the reordering of the candidate sentences provided by the recognizer as described in the following section.

3.4 The Combination Scheme

The proposed combination of the recognition score $\phi(W)$ defined in (3) with the probability $P_{SCFG}(W)$ provided by the parser introduced above is implemented through an additional weighted component and results in an extended sentence score $\psi(W)$ according to (5) below.

$$\psi(W) = \log P_{HMM}(X|W) + \alpha \log P_{BG}(W) + n\beta + \gamma \log P_{SCFG}(W). \quad (5)$$

Parameter γ will be called *Parse Scale Factor* (PSF) and weights the influence of the parse probability on the extended sentence score $\psi(W)$. For $\gamma = 0$, the sentence probability provided by the parser will not affect $\psi(W)$ at all. In this case, (5) and (3) will become identical. If $\gamma > 0$, the parse probability influences $\psi(W)$ and a reordering of the n-best candidate sentences may take place (see Fig. 3). Similarly to parameters α and β, which control the integration of the word bigram language model in the decoding

Rank	$\psi(W)$	$\phi(W)$	$P_{SCFG}(W)$	Candidate sentence
1	23,729	23,922	4.6286e-20	She has put up the value of her money .
2	23,703	23,924	7.6905e-23	She has put up the value other money .
3	23,700	23,888	1.5835e-19	She had put up the value of her money .
4	23,671	23,890	2.6311e-22	She had put up the value other money .
5	23,650	23,854	1.1246e-21	She has put up the value at her money .

Fig. 3. The reordered n-best list showing the resulting sentence scores $\psi(W)$ for $\gamma = 10$, the recognition scores $\phi(W)$, and the parse probabilities $P_{SCFG}(W)$. The original n-best list is the one shown in Fig. 2.

TABLE 1
The Definition of the Validation and Test Sets for the Multiwriter Task (MWT) and the Writer Independent Task (WIT)

	MWT		WIT	
	Validation	Test	Validation	Test
Writers	157	156	100	100
Sentences	200	200	200	200
Words	3,814	3,933	4,094	3,956
Lexicon size	8,824	8,822	8,827	8,821
SCFG productions	21,691	21,716	21,705	21,694

TABLE 2
Test Set Results for the Multiwriter (MWT) and the Writer Independent Task (WIT)

Task	Performance Measure	Baseline	Parsing	Significance
MWT	Sentence Recognition Rate	7.5%	8.0%	60%
	Word Recognition Rate	76.7%	77.2%	75%
	Word Level Accuracy	74.7%	75.6%	$> 99\%$
WIT	Sentence Recognition Rate	11.0%	12.0%	70%
	Word Recognition Rate	79.3%	79.4%	50%
	Word Level Accuracy	76.8%	77.6%	91%

process, parameter γ needs to be optimized experimentally on a validation set.

In the logarithmic space, the proposed integration of $P_{SCFG}(W)$ into $\psi(W)$ is equivalent to a linear combination of the effects of the word bigram language model and the SCFG. This combination scheme can also be interpreted as a mixture of experts. Instead of using an optical expert providing $P_{HMM}(X|W)$ and just a single language expert as in the case of (3), we now integrate two experts which cover different aspects of the underlying language. The value $P_{BG}(W)$ provided by the bigram language model is only based on directly adjacent words while $P_{SCFG}(W)$ evaluates the grammatical soundness of a complete sentence.

4 Experiments and Results

The proposed combination scheme of the baseline recognizer with the syntax analysis module is evaluated on a series of experiments. We first introduce the handwritten samples and linguistic resources involved in the experiments. Then, the experimental setup is explained and the obtained results are presented.

4.1 Handwritten Data and Linguistic Resources

All handwritten material, namely, images of handwritten English sentences are taken from the IAM database [23]. The database has been collected at the University of Bern to build, train, and test offline handwriting recognition systems for unconstrained English texts. Its automatic segmentation into individual words described in [33] also allows the extraction of text lines and complete sentences (see Fig. 1). The database now contains more than 1,500 scanned pages of handwritten text contributed by more than 600 different writers.

The text images provided with the IAM database are based on texts from the Lancaster-Oslo/Bergen (LOB) corpus [14] which contains 500 printed English texts of about 2,000 words each. To derive lexica, statistical language models, and the SCFG needed for syntax analysis we use the Tagged LOB (TLOB) [13] corpus and the Lancaster Parsed Corpus (LPC) [7]. The TLOB is based on the LOB corpus and contains its explicit segmentation into individual words. It further provides a grammatical tag for each word. The LPC is a treebank containing the parse trees of 11,827 sentences selected from the LOB corpus.

4.2 Experimental Setup and System Optimization

Two different recognition tasks are defined. In the *Multiwriter Task* (MWT) the recognizer is trained on handwritten texts from a large set of known writers. For the *Writer Independent Task* (WIT), writing styles are not known in advance, i.e., the writers represented in the training set are not represented in either the validation or the test set of this task. For the training of the recognizer 5,799 handwritten text lines written by 448 different persons have been selected from the IAM database. This training set supports both the MWT and the WIT task at the same time. The validation sets are used to find optimal values of system parameters while the system performance is evaluated on the test sets (see Table 1). In our experimental setup, we assume that each handwritten input is a proper English sentence. Furthermore, we assume that all words occurring in an input sentence are included in the vocabulary.

For the performance evaluation we use the sentence recognition rate, the word recognition rate, and the word level accuracy. The sentence recognition rate measures the percentage of correctly recognized sentences where a sentence is considered to be correctly recognized if and only if the recognition result matches its transcription (ground truth) word by word. The different possible types of errors are called substitutions (S), insertions (I), and deletions (D) where each misrecognized word leads to a substitution error. If the recognizer erroneously splits a single word into two parts, an insertion error is generated. Missed spaces between two consecutive words lead to deletion errors. The word recognition rate measures the fraction of correctly recognized words and is defined by $(N - D - S)/N$, where N is the total number of words in the transcription of a sentence. The word level accuracy $(N - D - S - I)/N$ also takes insertion errors into account. It is therefore a more appropriate measure of the quality of the recognition result.

After the initial training of the HMM-based recognition system using and the Baum-Welch algorithm [26], the integration of the word bigram language model and the Parse Scale Factor (PSF) γ were optimized on the validation sets, according to (5). For the tuning of the PSF, an exhaustive search over the parameter space from $\gamma = 0$ to $\gamma = 20$ was applied leading to $\gamma = 10$ for the WIT and $\gamma = 13$ for the MWT. For grammatically incorrect sentences (i.e., the parser did not find a parse tree for the given sentence), a fixed minimum parse probability of 10^{-300} was assumed. This simple scheme resulted in identical recognition rates on the validation data as another more elaborate thresholding method which took into account the parse probabilities for the n-best list of candidates sentences. The value of the fixed minimum parse probability has been determined on the validation sets. Please note that this minimum parse probability is effectively working as a filter which always favors grammatically correct sentences over grammatically incorrect solutions.

4.3 Test Set Results

The final results obtained on the test sets for the MWT and the WIT are summarized in Table 2. Column "Baseline" contains the results of the baseline recognizer and column "Parsing" holds the corresponding results for the combined system including the syntax analysis module. In the last column, the significance of the improvement is reported which is computed using the correlated Z-test. This test allows to compute the probability that the measured improvements are not just produced by chance. The highest significance of 99 percent is reached for the increase of the word level accuracy from 74.7 percent to 75.6 percent (+0.9 percent) of the MWT. Although these results are not very impressive at first glance, they compare favorably with the best published results in the domain of speech recognition for broad coverage grammars.

According to [22], the use of a word bigram language model leads to a substantial improvement of the recognition rate. Hence, it seems to be difficult to further boost the performance by means of syntax analysis. To confirm this hypothesis, we also measured the performance without language model and without syntax analysis. In this case, a word level accuracy of 49.1 percent was obtained on the WIT test set. Next, the SCFG-based syntax analysis module was added (without bigram language model). This led to an improvement from 49.1 percent to 54.4 percent. We therefore conclude that the SCFG-based syntax analysis procedure proposed

in this paper has the potential of substantially increasing the performance of a recognizer. However, this improvement becomes smaller for already intensively optimized recognizers.

5 Conclusion

We have proposed a combination scheme for an HMM-based offline handwritten sentence recognizer and a syntax analysis module which includes parsing of English sentences using a broad coverage SCFG. The main goals of the syntax analysis module are to improve recognition performance by penalizing grammatically unlikely candidate sentences and to provide additional linguistic information which could be used in other contexts as semantic information retrieval or text understanding.

After carefully optimizing both the baseline recognizer and the proposed combination with the syntax analysis module, improvements of the word level accuracy of around 1 percent (absolute) were achieved. These results compare favorably with the results published in the domain of speech recognition for the use of such grammars. Since these results are achieved using a large broad coverage grammar for written English, almost no constraints are imposed on the handwritten texts to be recognized. Furthermore, the proposed combination scheme requires only a loose coupling of the recognizer and the syntax analysis module. It is therefore simple to implement and to test.

Future research could include open vocabulary recognition and comparison of recognition rates resulting from the use of different grammars. Such grammars could either be extracted from additional parsed corpora or they could be directly inferred from large amounts of text.

Acknowledgments

This research was partly supported by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management" (IM2) in the individual Project "Scene Analysis."

References

[1] J.-C. Chappelier and M. Rajman, "A Generalized CYK Algorithm for Parsing Stochastic CFG," *Actes de TAPD*, pp. 133-137, 1998.

[2] J.-C. Chappelier, M. Rajman, R. Aragüés, and A. Rozenknop, "Lattice Parsing for Speech Recognition," *Proc. Sixth Conf. sur le Traitement Automatic du Langage Naturel*, pp. 95-104, 1999.

[3] C. Chelba and F. Jelinek, "Structured Language Modeling," *Computer Speech and Language*, vol. 14, pp. 283-332, 2000.

[4] C. Crowner and J. Hull, "A Hierarchical Pattern Matching Parser and Its Application to Word Shape Recognition," *Proc. First Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 323-331, 1991.

[5] G. Erbach, "Bottom-Up Earley Deduction," *Proc. 14th Int'l Conf. Computational Linguistics*, 1994.

[6] J. García-Hernandez, J.-A. Sánchez, and J.-M. Benedí, "Performance and Improvements of a Language Model Based on Stochastic Context-Free Grammars," *Proc. 11th Iberian Conf. Pattern Recognition and Image Analysis*, pp. 271-278, 2003.

[7] R. Garside, G. Leech, and T. Váradi, *Manual of Information for the Lancaster Parsed Corpus*. Bergen, Norway: Norwegian Computing Center for the Humanities, 1995.

[8] I. Good, "The Population Frequencies of Species and the Estimation of Population Parameters," *Biometrika*, vol. 40, pp. 237-264, 1953.

[9] N. Gorski, V. Anisimov, E. Augustin, D. Price, and J.-C. Simon, "A2iA Check Reader: A Family of Bank Check Recognition Systems," *Proc. Fifth Int'l Conf. Document Analysis and Recognition*, pp. 523-526, 1999.

[10] S.L. Graham, M.A. Harrison, and W.L. Ruzzo, "An Improved Context-Free Recognizer," *ACM Trans. Programming Languages and Systems*, vol. 2, no. 3, pp. 415-462, 1980.

[11] T. Hong and J. Hull, "Text Recognition Enhancement with a Probabilistic Lattice Chart Parser," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 222-225, 1993.

[12] J. Hull, "Incorporating Language Syntax in Visual Text Recognition with Statistical Model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1251-1256, Dec. 1996.

[13] S. Johansson, E. Atwell, R. Garside, and G. Leech, *The Tagged LOB Corpus, Users's Manual*. Bergen, Norway: Norwegian Computing Center for the Humanities, 1986.

[14] 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," Dept. of English, Univ. of Oslo, Norway, 1978.

[15] D. Jurafsky, C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchman, and N. Morgan, "Using a Stochastic Context-Free Grammar as a Language Model for Speech Recognition," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 189-192, 1995.

[16] S. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400-401, 1987.

[17] F. Keenan, L. Evett, and R. Whitrow, "A Large Vocabulary Stochastic Syntax Analyser for Handwriting Recognition," *Proc. First Int'l Conf. Document Analysis and Recognition*, pp. 794-802, 1991.

[18] K. Kita, T. Kawabata, and H. Saito, "HMM Continuous Speech Recognition Using Predictive LR Parsing," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 703-706, 1989.

[19] K. Kita and W. Ward, "Incorporating LR Parsing into SPHINX," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 269-272, 1991.

[20] A.L. Koerich, Y. Leydier, R. Sabourin, and C.Y. Suen, "A Hybrid Large Vocabulary Handwritten Word Recognition System Using Neural Networks and Hidden Markov Models," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition*, pp. 99-104, 2002.

[21] U. Mahadevan and S.N. Srihari, "Parsing and Recognition of City, State, and Zipcodes in Handwritten Addresses," *Proc. Fifth Int'l Conf. Document Analysis and Recognition*, pp. 325-328, 1999.

[22] U.-V. Marti and H. Bunke, "Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 15, pp. 65-90, 2001.

[23] U.-V. Marti and H. Bunke, "The IAM-Database: An English Sentence Database for Off-Line Handwriting Recognition," *Int'l J. Document Analysis and Recognition*, vol. 5, pp. 39-46, 2002.

[24] A. Ogawa, K. Takeda, and F. Itakura, "Balancing Acoustic and Linguistic Probabilities," *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing*, pp. 181-184, 1998.

[25] P.Y. Simard, D. Steinkraus, and J. Platt, "Best Practices for Convolutional Nerual Networks Applied to Visual Document Analysis," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, vol. 2, pp. 958-962, 2003.

[26] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[27] B. Roark, "Probabilistic Top-Down Parsing and Language Modeling," *Computational Linguistics*, vol. 27, no. 2, pp. 249-276, 2001.

[28] R. Srihari, S. Ng, C. Baltus, and J. Kud, "Use of Language Models in On-Line Sentence/Phrase Recognition," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition*, pp. 284-294, 1993.

[29] S. Uchida and H. Sakoe, "An Off-Line Character Recognition Method Employing Model-Dependent Pattern Normalization by an Elastic Membrane Model," *Proc. Fifth Int'l Conf. Document Analysis and Recognition*, pp. 499-502, 1999.

[30] A. Vinciarelli, S. Bengio, and H. Bunke, "Offline Recognition of Unconstrained Handwritten Texts Using HMM and Statistical Language Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 709-720, June 2004.

[31] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Information Theory*, vol. 13, no. 2, pp. 260-269, 1967.

[32] F. Voisin and J. Raoult, "A New, Bottom-Up, General Parsing Algorithm," *Journées AFCET-GROPLAN, les Avancées en Programmation*, 1990.

[33] M. Zimmermann and H. Bunke, "Automatic Segmentation of the IAM Off-Line Handwritten English Text Database," *Proc. 16th Int'l Conf. Pattern Recognition*, vol. 4, pp. 35-39, 2002.

[34] M. Zimmermann and H. Bunke, "Hidden Markov Model Length Optimization for Handwriting Recognition Systems," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition*, pp. 369-374, 2002.

[35] M. Zimmermann and H. Bunke, "Optimizing the Integration of Statistical Language Models in HMM Based Offline Handwritten Text Recognition," *Proc. 17th Int'l Conf. Pattern Recognition*, vol. 2, pp. 541-544, 2004.

[36] M. Zimmermann, J.-C. Chappelier, and H. Bunke, "Parsing n-best Lists of Handwritten Sentences," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 572-576, 2003.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.