

N-Gram Language Models for Offline Handwritten Text Recognition

Matthias Zimmermann and Horst Bunke
Department of Computer Science, University of Bern
Neubrückestrasse 10, CH-3012 Bern, Switzerland
{zimmerma,bunke}@iam.unibe.ch

Abstract

This paper investigates the impact of bigram and trigram language models on the performance of a Hidden Markov Model (HMM) based offline recognition system for handwritten sentences. The language models are trained on the LOB corpus which is supplemented by various additional sources of text, including sentences from additional corpora and random sentences produced by a stochastic context-free grammar (SCFG). Experimental results are provided in terms of test set perplexity and performance of the corresponding recognition systems. For the text recognition experiments handwritten material from the IAM database has been used.

1 Introduction

Offline recognition of general handwritten text has been investigated since many years [2, 13, 17]. Only recently authors started to take advantage of the integration of *n*-gram language models to improve the accuracy of the corresponding recognizers. Today, *n*-gram modeling techniques are well established and represent the most frequently applied language model in today's speech recognition systems [16]. Furthermore, toolkit support is available for training and evaluation of such models [4] as well as for their integration into HMM based recognition systems [19]. The integration of word bigram language models into HMM based handwritten text recognition systems has been reported in [14, 18] while word trigram models were used in [18].

In this paper we address training and use of word bigram and word trigram language models in the context of handwritten sentence recognition. Specifically, we compare natural sentences and random sentences generated by a stochastic context-free grammar (SCFG) as additional sources of text for the training of the statistical language models.

The remaining sections are organized as follows. Sec. 2 reviews related work and Sec. 3 describes the applied methodology. Experiments and results are reported in Sec. 4, and conclusions are drawn in Sec. 5.

2 Related Work

The use of word bigram language models in HMM based handwriting recognition systems proved to be very effective [8, 14, 18], while no further improvement was observed using word trigram language models in [18]. Since millions of words are required for the training of both bigram and trigram language models, large corpora are needed to estimate the parameters of these models. As an alternative to natural text, the use of random sentences generated by a SCFG has been proposed in [11]. In the following paragraphs experiments and results as reported in [8, 11, 14, 18] are briefly described.

A word tag bigram language model was used in [8]. Instead of relying on word transition probabilities the implemented model bases on the combination of syntactical word tag transition statistics and word probabilities. Although no backoff or smoothing technique was used, an increase of the word recognition rate from 51% to 61% has been reported for the chosen experimental setup.

The successful use of synthetic data for the training of a bigram language model has been reported in the context of the Berkeley Restaurant Project in the domain of speech recognition [11]. First, a pseudo-corpus has been compiled by using a SCFG in generation mode to produce 200,000 random sentences. This pseudo-corpus was then added to the initial corpus containing 4,786 sentences to train a new bigram language model. In an experiment using 364 test sentences randomly selected from the regular corpus the performance of the two recognition systems was compared. As a result, the system using the new bigram language model was able to reduce the word error rate from 34.6% to 29.6%.

In [14] unigram and bigram language models for medium sized lexicons are investigated. Making the closed lexicon assumption, which means that all words from the test set are included in the lexicon, recognition rates of both unigram and bigram based recognizers are compared. For different sizes of the lexicon (2,700-7,700 words) the bigram based recognizer produced word recognition rates which were between 9% and 18% higher than the rates of the corresponding unigram based recognizers.

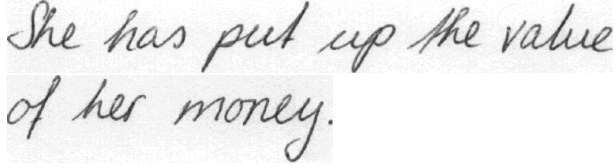


Figure 1. A sample sentence from the IAM Database.

Trigram and bigram language models for medium to large lexicons (5,000-50,000 words) have been used in [18] to study the recognition performance of an offline handwritten text line recognition system. In correspondence to the results published in [14] the system using the bigram language model outperformed the system using a unigram model by 10% in average for large lexicons. However, no further improvement was achieved using trigram language models. In order to explain this counter intuitive findings the authors suggest two possible reasons. First, the individual recognition of handwritten lines of text restricts the possible benefit of higher order n -gram models (e.g. the full power of trigram model will only be available after the second word of each line). Second, the TDT-2 newswire corpus [3], which was used to train the bigram and the trigram language models, is not aligned with the actual handwritten texts based on material from the LOB corpus [10].

3 Methodology

Using an HMM based handwritten text recognition system, handwritten sentences (see Fig. 1 for an example) are recognized line by line. For the recognition of a handwritten text line we are interested in finding the most likely sentence $\hat{W} = (w_1, w_2 \dots w_n)$ for a given observation sequence $X = (X_1, X_2, \dots X_m)$ provided by the recognizers' feature extraction mechanism. The text \hat{W} is found according to Eq. (1), where the sentence probability term $p(W)$ is provided by a statistical language model.

$$\hat{W} = \underset{W}{\operatorname{argmax}} p(X|W)p(W) \quad (1)$$

Instead of just obtaining the most likely candidate sentences, recognition lattices are produced for the experiments reported in this paper. A recognition lattice can be interpreted as a directed acyclic graph. Such a graph represents the most promising part of the search space investigated during the decoding process [20, 23]. The nodes of a lattice represent boundaries between candidate words which are associated with potential segmentation points of the input sequence X . The candidate words are attached to the edges of a recognition lattice and the corresponding recognition scores are computed using the following formula.

$$\log p(X_i|w_i) + \alpha \log p(w_i|w_1^{i-1}) + \beta \quad (2)$$

where X_i represents the feature vector sequence associated with word w_i . The score $p(X_i|w_i)$ is calculated by the optical model (the HMM) and the value $p(w_i|w_1^{i-1})$ ¹ is provided by a language model. Since both the HMM and the language model only produce approximations of the true probabilities we use two additional parameters, α and β . Their aim is to partially compensate the deficiencies of the optical model and the language model. Optimal values for α and β are then determined by experiment on a validation set [23].

In the following subsections n -gram language modeling is introduced first. Then generation of random sentences is described.

3.1 N -Gram Language Modeling

N -gram language models provide a simple approximation for sentence probabilities based on the relative frequencies of word sequences of length n . For $n = 2$ ($n = 3$) we use the term bigram (trigram) language model. The sentence probability $p(s)$ is decomposed into a product of conditional probabilities $p(w_i|w_{i-n+1}^{i-1})$ ² as follows

$$p(s) = \prod_{i=1} p(w_i|w_{i-n+1}^{i-1}) \quad (3)$$

The n -gram probabilities $p(w_i|w_{i-n+1}^{i-1})$ are estimated using the relative frequencies $f(w_i|w_{i-n+1}^{i-1})$ of the corresponding word n -grams found in large training corpora. To assign positive probabilities to n -grams which have not been observed in the training text model smoothing is applied. For the experiments reported in this paper the Good-Turing smoothing technique [7] together with a backoff to lower order models [12] is used.

In order to evaluate the quality of a statistical language model the perplexity is the most frequently used measure. For a given language model M and test text $T = (s_1, s_2, \dots s_n)$ it is defined using Eq. (4) where $s_i = (w_1, \dots, w_{n_i})$ stands for the i^{th} sentence of the test text.

$$PP_T(M) = 2^{H_T(M)} \quad (4)$$

where $H_T(M)$ represents the (cross)entropy of the language model M for text T . The entropy $H_T(M)$ can be estimated using the sentence probabilities provided by the language model M as shown below.

$$H_T(M) = -\frac{1}{n} \sum_{i=1}^n \log p(s_i) \quad (5)$$

The perplexity can intuitively be interpreted as the average number of relevant words in the lexicon. Language models with lower perplexity values are therefore better suited to

¹ The term w_1^{i-1} corresponds to the word history $(w_1, w_2, \dots, w_{i-1})$.

² w_{i-n+1}^{i-1} corresponds to the truncated word history of length n , i.e. $w_{i-n+1}^{i-1} = (w_{i-n+1}, \dots, w_{i-1})$.

'explain' a test text than models with higher perplexity values.

3.2 Generation of Random Sentences

Arbitrary amounts of (grammatically correct) random sentences can be produced using a Stochastic Context-Free Grammar (SCFG) in generation mode. For this paper the SCFG corresponds to a treebank grammar which is extracted from the Lancaster Parsed Corpus [6]. After a brief definition of SCFG and the presentation of the extraction of a treebank grammar the generation of a sample sentence is described.

A SCFG is a 5-tuple $(N, T, P, S, p(\cdot))$ where N represent the set of nonterminal symbols and T the set of terminal symbols, $N \cap T = \emptyset$. The set of productions is denoted by P and $S \in N$ is used as the start symbol. All productions of P can be written as $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup T)^+$. Finally, the probability function $p(\cdot)$ maps productions $A \rightarrow \alpha$ into the interval $(0, 1]$ where the probabilities of all $A \rightarrow \alpha_i \in P$ have to satisfy $\sum_{\alpha} p(A \rightarrow \alpha) = 1$.

Treebank grammars can be extracted from corresponding corpora (also called treebanks) which contain parsed sentences in the form of hierarchical derivation trees (see Fig.2 for an example).

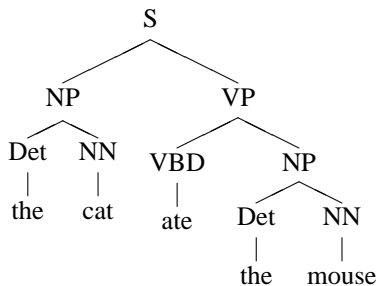


Figure 2. A parsed sentence

Using the available derivation trees it is straightforward to extract corresponding productions. The production probabilities can then be estimated from the relative frequencies as follows.

$$p(A \rightarrow \alpha) = \frac{N(A \rightarrow \alpha)}{\sum_{\beta} N(A \rightarrow \beta)} \quad (6)$$

where $N(A \rightarrow \alpha)$ represents the number of times the production $A \rightarrow \alpha$ has been observed in the treebank. This count is then normalized by the sum of the counts of all productions with the same left hand side A . Applying this scheme to the parsed sentence from Fig. 2 the SCFG shown in Fig. 3 is obtained.

For the generation of a random sentence a rewriting process is applied in the following way. First, the start symbol S is rewritten into a string of symbols α by a production $S \rightarrow \alpha$. From the resulting string a nonterminal symbol is

(1.0) S	→	NP VP	(0.5) NN	→	mouse
(1.0) NP	→	Det NN	(0.5) NN	→	cat
(1.0) VP	→	VBD NP	(1.0) VBD	→	ate
(1.0) Det	→	the			

Figure 3. Productions of a stochastic context-free grammar for a subset of English sentences

selected and rewritten using a suitable production. If several alternative productions are available, a production is selected randomly according to its probability. This procedure terminates when no more nonterminal symbols are found in the resulting string.

$S \Rightarrow NP VP \Rightarrow Det NN VP \Rightarrow the NN VP \Rightarrow the mouse VP \Rightarrow the mouse VBD NP \Rightarrow \dots \Rightarrow the mouse ate the cat$

Figure 4. Generation of a random sentence

Using the production scheme described above it cannot be guaranteed that the sentences are semantically meaningful and the test set may even contain grammatically incorrect sentences which cannot be generated by the SCFG. Therefore it is not advisable to extract n -gram language models directly from random sentences. Instead, random sentences should be added to an initial corpus of natural text before an n -gram language model is extracted from the combined set of sentences.

4 Experiments and Results

The data used for the experiments can be divided into handwritten material and linguistic resources. For the training of the character HMM and the recognition of the sentences, images of handwritten lines of text and complete sentences were automatically extracted from the segmented version of the IAM database [15, 21].

Lexica, bigram and trigram language models for the baseline recognizer were extracted from the tagged LOB corpus [9]. A variable number of additional sentences were taken from both the Brown corpus [5] and the Wellington corpus [1]. All three corpora contain approximately 1,000,000 running words each and cover a similar variety of texts chosen for their representative quality of written English.

The treebank grammar for the generation of the random sentences has been extracted from the Lancaster Parsed Corpus (LPC) [6] which contains parse trees for 10,000 sentences from the LOB corpus.

For the experimental setup a writer independent environment, where the set of writers who contributed to the training, test, and validation set are mutually disjoint was chosen. The training set consists of 5,799 handwritten text lines in-

cluding 39,993 word instances written by 448 different persons. Both validation and test set consist of 200 complete sentences each (average length 20 words), written by 100 writers. The task lexicon is closed³ over the test (validation) set and includes 8,819 (8,825) words. For the extraction of the baseline bigram and trigram language model all sentences from the test (validation) set were excluded from the tagged LOB Corpus.

The HMM based handwritten text recognition system described in [14] is using a linear topology for the character models. This topology was adopted in the current paper. However, the number of states was chosen depending on the individual character [22], and a mixture of eight Gaussians for each state was used, rather than just a single Gaussian as reported in [14].

4.1 Optimizing Perplexity

The first set of experiments investigates the effect of additional training material on the validation set perplexity. Four different sources of additional sentences are investigated to enhance the language models trained on the tagged LOB corpus: random sentences produced by the treebank grammar as well as natural sentences from the Brown corpus, the Wellington corpus and, the combined Brown + Wellington corpus⁴.

The results for the language models including an increasing number of sentences from the different sources of additional text are provided in Fig. 5. For all types of additional text the perplexity values achieved by the trigram models outperformed those obtained by the corresponding bigram models by approximately 20%. The performance of the random sentences is discouraging. An almost linear increase in perplexity was obtained adding random sentences to the LOB corpus. While the incorporation of the complete Brown corpus reduced the perplexity by 6% for trigram model, a reduction of 12% was measured when the complete Wellington corpus has been added to the LOB corpus. The combined Brown and Wellington corpus lead to a perplexity reduction of 15%.

4.2 Optimizing System Performance

The goal of the second set of experiments is to determine the language model which maximizes the recognition performance. Therefore all language models produced for the perplexity experiments are used to rescore the recognition lattices obtained for the 200 sentences of the validation set. Fig. 6 provides the measured word level accuracies⁵ of the corresponding recognition systems.

³ The closing the lexicon over the test (validation) set ensures that all words of the test (validation) set are contained in the task lexicon.

⁴ For the construction of the combined corpus the sentences were taken from the Brown and the Wellington corpus in an alternating way.

⁵ The word level accuracy is defined as $(H - I)/N$ where N represents the number of words in the correct solution, H stands for the number of correctly recognized words and the number of insertions is specified by I .

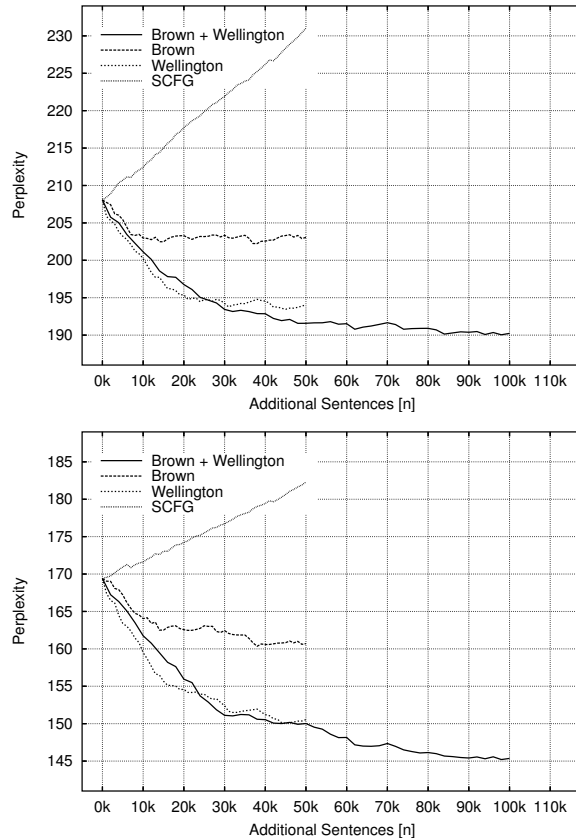


Figure 5. Validation set perplexity for different bigram (top) and trigram language models (bottom)

If only the LOB corpus is used for training, rescoring with the trigram language model achieves a word level accuracy almost 0.5% higher than rescoring with the baseline bigram model. In the case where additional sentences are used, trigram model performance is increased by 2% while the increase of the bigram language model is 1% only. The use of random sentences clearly decreases the performance of the bigram language model systems. The impact of the random sentences on the trigram language model systems is unclear. As can be seen in Fig. 6, the best validation set performance has been measured for the trigram language model trained on the sentences from the LOB corpus supplemented by the first 65,000 sentences of the combined Brown + Wellington corpus.

4.3 Test Set Results

For the first test set experiment the performance of the baseline trigram model trained on the tagged LOB corpus only is compared to the previously used bigram language model trained on the same data.

The results from this experiment are summarized in Tab. 1.

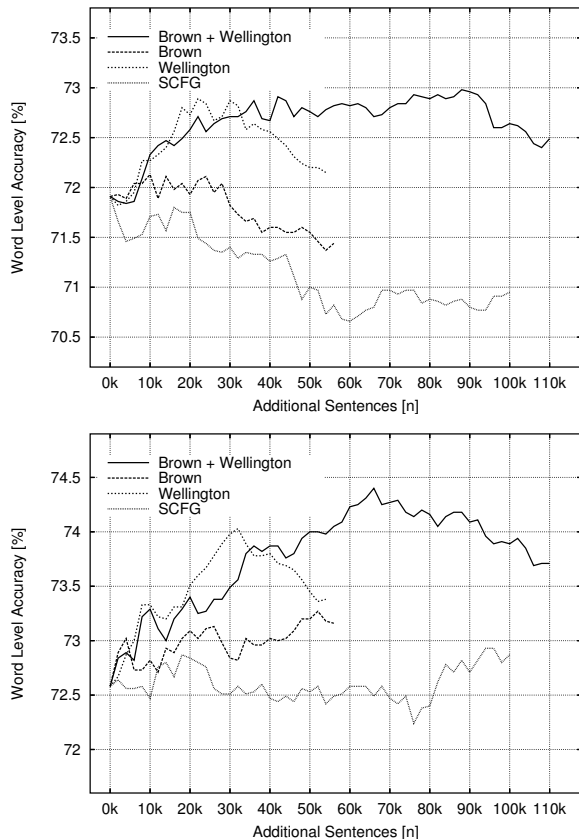


Figure 6. Word level accuracies for different different bigram (top) and trigram language models (bottom) on the validation set

For each performance measure the corresponding rates of the baseline bigram and the baseline trigram language model are compared⁶.

In a second experiment the configuration of the language model which showed the best performance on the validation set is used to produce a corresponding trigram model. Tab. 1 provides the comparison of the system using the baseline trigram model (column 'Trigram') extracted from the LOB corpus only and the system using the smoothed trigram language model (column 'S. Trigram'). Except for the increase of the sentence recognition rate from 11% to 12% all achieved improvements over the baseline system using either the baseline bigram or the baseline trigram language model were found to be significant on the 95% level.

The n -best analysis shown in Fig. 7 provides a comparison of the baseline system integrating the bigram language model (trained on the LOB corpus only) and the system using the

⁶ The sentence recognition rate measures the percentage of the correctly recognized sentences and the word recognition rate is defined as H/N . Please note that no better sentence recognition rates can be expected given the average length of the sentences (20 words).

Perf. Measure	Bigram	Trigram	S. Trigram
Sen. Rec. Rate	11.0%	12.0%	14.0%
Word Rec. Rate	79.0%	80.3%	81.8%
Word Level Acc.	76.3%	78.2%	79.9%

Table 1. Test set results for the baseline trigram model and the smoothed trigram model

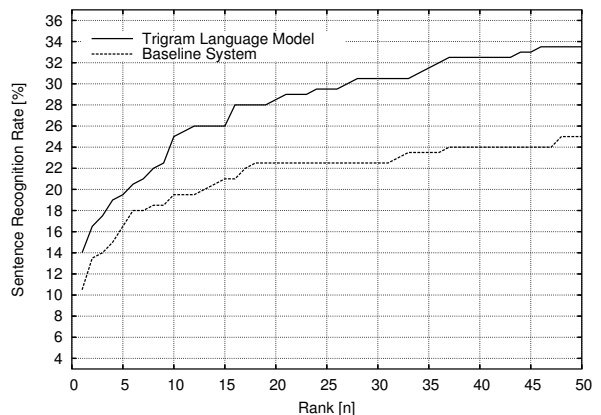


Figure 7. N -best analysis of the test set sentence recognition rate.

smoothed trigram language model. The figure shows that the resulting increase of the sentence recognition rate is getting bigger as more additional ranks are included in the analysis.

5 Conclusions

This paper investigated bigram and trigram language models in the context of offline handwritten text recognition using different types and amounts of training texts. Specifically, the use of random sentences produced by a SCFG to supplement the LOB corpus to train both bigram and trigram language models has been addressed.

The results of the experiments based on the random sentences generated by the SCFG did not bring any improvements over the baseline system. This finding is in contrast to the results published in [11] where a similar pseudo-corpus was successfully used to smooth a bigram grammar in the context of the Berkeley Restaurant Project. The discrepancy between these results may be explained by the fact that a rather small and task-specific SCFG was used for the experiments published in [11]. It can be assumed that such grammars generate random sentences better aligned with the recognition task at hand than the large general-purpose tree-bank grammar used in this paper.

The trigram language models significantly outperformed the bigram language models used previously. These findings do not correspond to [18] where no improvement of the

system using a trigram language model over the baseline bigram language model was observed. The following two reasons are believed to explain the different findings. First, the lines of handwritten text recognized by the system described in [18] contain roughly 10 words on the average. Since a trigram model starts to be helpful only after the third word only 80% of the words in a text line could benefit from such language models. Second, the linguistic resources do not seem to be as well aligned in [18] as in the experimental setup chosen for this paper.

The experimental results presented in this paper clearly indicate that word trigram language models can further improve the recognition performance of a handwritten recognition system based on word bigram language model. To take full advantage of the trigram language models we recommend that the recognition results of several lines of handwritten text should be concatenated and suggest that the training texts should be well aligned with the test texts represented by the handwritten material.

References

- [1] L. Bauer. *Manual of Information to accompany The Wellington Corpus of Written New Zealand English, for use with Digital Computers*. Department of Linguistics, Victoria University, Wellington, New Zealand, 1993.
- [2] R. Bozinovic and S. Srihari. Off-line cursive script word recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(1):68–83, Jan. 1989.
- [3] C. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel. The TDT-2 text and speech corpus. In *DARPA Broadcast News Workshop*, 1999.
- [4] P. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. <http://mi.eng.cam.ac.uk/prc14/toolkit.html>. In *5th Europ. Conf. on Speech Communication and Technology, Rhodes, Greece*, volume 5, pages 2707 – 2710, 1997.
- [5] W. N. Francis and H. Kucera. *Brown Corpus Manual, Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers*. Department of Linguistics, Brown University, Providence RI, USA, 1979.
- [6] R. Garside, G. Leech, and T. Váradi. *Manual of Information for the Lancaster Parsed Corpus*. Norwegian Computing Center for the Humanities, Bergen, 1995.
- [7] I. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.
- [8] V. Govindaraju, R. Srihari, and S. Srihari. Handwritten text recognition. In A. L. Spitz and A. Dengel, editors, *Document Analysis Systems*, pages 288–304. World Scientific, 1995.
- [9] S. Johansson, E. Atwell, R. Garside, and G. Leech. *The Tagged LOB Corpus, Users’s Manual*. Norwegian Computing Center for the Humanities, Bergen, Norway, 1986.
- [10] 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*. Department of English, University of Oslo, Oslo, 1978.
- [11] 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. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 189–192, Detroit MI, USA, 1995.
- [12] S. 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(3):400–401, 1987.
- [13] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. An unconstrained handwriting recognition system. *Int. Journal on Document Analysis and Recognition*, 4:226–242, 2002.
- [14] 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*, 15:65–90, 2001.
- [15] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for off-line handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.
- [16] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proc. of the IEEE*, 88:1270–1278, 2000.
- [17] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):309–321, Mar. 1998.
- [18] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of large vocabulary cursive handwritten text. In *7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland*, volume 2, pages 1101–1105, 2003.
- [19] S. J. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, editors. *The HTK Book (for HTK Version 3.2)*. <http://htk.eng.cam.ac.uk/>. Cambridge University Engineering Department, 2002.
- [20] S. J. Young, N. H. Russell, and J. H. S. Thornton. Token passing: a conceptual model for connected speech recognition systems. CUED technical report F INFENG/TR38, Cambridge University, 1989.
- [21] M. Zimmermann and H. Bunke. Automatic segmentation of the IAM off-line handwritten English text database. In *16th Int. Conf. on Pattern Recognition*, volume 4, pages 35–39, Quebec, Canada, Aug. 2002.
- [22] M. Zimmermann and H. Bunke. Hidden Markov model length optimization for handwriting recognition systems. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 369–374, Niagra-on-the-Lake, Canada, Aug. 2002.
- [23] M. Zimmermann and H. Bunke. Optimizing the integration of statistical language models in HMM based offline handwritten text recognition. In *Accepted for 17th Int. Conf. on Pattern Recognition*, Cambridge, England, 2004.