On the Influence of Vocabulary Size and Language Models in Unconstrained Handwritten Text Recognition

U.-V. Marti and H. Bunke
Institut für Informatik und angewandte Mathematik
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
email:{marti,bunke}@iam.unibe.ch

Abstract

In this paper we present a system for unconstrained handwritten text recognition. The system consists of three components: preprocessing, feature extraction and recognition. In the preprocessing phase, a page of handwritten text is divided into its lines and the writing is normalized by means of skew and slant correction, positioning and scaling. From a normalized text line image, features are extracted using a sliding window technique. From each position of the window nine geometrical features are computed. The core of the system, the recognizer, is based on hidden Markov models. For each individual character, a model is provided. The character models are concatenated to words using a vocabulary. Moreover, the word models are concatenated to models that represent full lines of text. Thus the difficult problem of segmenting a line of text into its individual words can be overcome. To enhance the recognition capabilities of the system, a statistical language model is integrated into the hidden Markov model framework. To preselect useful language models and compare them, perplexity is used. Both perplexity as originally proposed and normalized perplexity are considered.

In our experiments several system configurations with different vocabulary sizes were tested. While the perplexity increases with a growing vocabulary, we observed that the normalized perplexity decreases. This leads to the conclusion that language models become more powerful in recognition tasks with larger vocabulary size. This conclusion could be confirmed in a number of experiments. For a system based on a vocabulary of 412 words a word recognition rate of 78.53% was measured when no language model was engaged. Using a bigram language model, the recognition rate incresed to 81.27%. For a 7719 word vocabulary, 40.47% of the words were recognized correctly without a language model, and 60.05% with bigram information.

Keywords: *handwriting recognition, unconstrained text recognition, hidden Markov models, statistical language modelling, perplexity, vocabulary size.*

1 Introduction

During the last years handwriting recognition has become an intensive research topic. While the first systems read segmented characters [1], later efforts aimed at the recognition of cursively handwritten words [2]. Only a short time ago the first systems appeared which are able to read sequences of words. Examples are systems which are able to recognized handwritten check amounts [3, 4] or postal addresses [5]. But these applications still operate in small, very specific domains.

At the moment only very few systems are known which address the domain of free text recognition [6, 7]. Typically, these systems segment the text into single words during a preprocessing phase. But as it is known that the segmentation of complete sentences into single words is difficult and prone to errors [8]. In particular, in the recognition phase it is hard to recover from errors that occurred during the earlier stage of segmentation.

In the present paper we propose a system that treats complete handwritten lines of text as basic input units. Segmentation of a line of text into individual words is obtained as a byproduct of recognition, which is based on hidden Markov models (HMMs). Another novel feature of the proposed system is the use of a statistical language model, which is incorporated in the recognizer to improve its performance. In an experimental evaluation the recognition performance of the system was tested. Particular attention was paid to measuring the recognition rate depending on the vocabulary size and the amount of information incorporated in the language model. The recognition rate obtained in the experiments was also related to the perplexity of the underlying language model, which can be analytically computed. Various versions of our system were trained and tested on the database described in [9]. This database consists of a col-

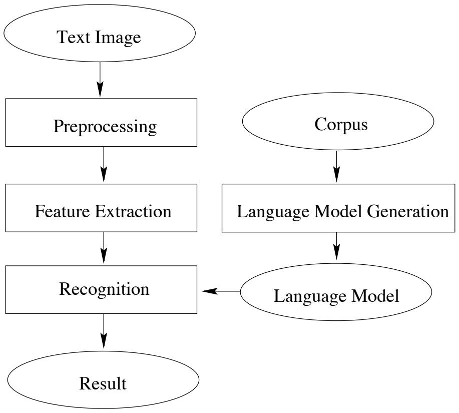

Figure 1. System overview.

lection of 9157 lines of text written by about 400 writers. The total number of word instances in the database is 82227 words.

In the next section we show how the data are preprocessed. The features selected for recognition are described in Section 3. In Section 4 the recognition procedure, the underlying language models and the perplexity are introduced. The experimental results obtained with this system are presented in Section 5. At the end in Section 6 we draw some conclusions from our work.

2 Preprocessing

The system described in this paper consists of three major parts that perform preprocessing, feature extraction and recognition (see Fig 1). In this section the preprocessing procedures are described.

Whole pages of text are used for training and testing the system. First, a page of text is segmented into individual lines. Then normalization operations are applied, including skew and slant correction, positioning and scaling.

To split a given text into text lines, for each row the black/white-transitions are counted. To eliminate outliers, the values are smoothed with a median filter. In the resulting histogram local minima are determined. If the value at a local minimum is zero, a horizontal cut has been found that doesn't touch any word. If the value is greater than zero, we have found a position where we can horizontally cut the image with a minimal number of intersections with strokes belonging to words of the previous or the following text line. To handle intersections of this kind a method based on the center of gravity is used. If the center of gravity of the connected component that is cut is in the range of the previous (the following) text line, the connected component is assumed to belong to that text line. If the center of gravity is near the cutting line, the component is cut into two parts, one belonging to the previous and one to the following text line. An example is shown in Fig. 2.

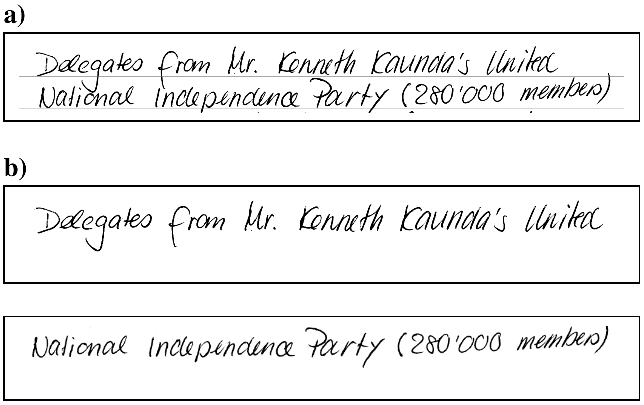
a)
Delegates from Mr. Kenneth Kaunda's United
National Independence Party (280'000 members)
b)
Delegates from Mr. Kenneth Kaunda's United
National Independence Party (280'000 members)

Figure 2. a) Two lines of text that can's be separated without a cut. b) Result of the line segmentation (see text).

Once the text lines are extracted, the skew of the writing in each line is corrected. For this purpose the lowest black pixel in each column is determined. These points are approximated by a straight line using regression analysis. Outliers, such as contour points resulting from descenders, are eliminated by using a threshold that defines a maximal error for regression.

To determine slant, the writing's contour is approximated by straight lines. In a histogram the angles weighted by each line's length, are accumulated. It is assumed that the maximum value in this histogram represents the slant angle.

For positioning and vertical scaling the baselines of the text lines are computed. This is done by fitting an ideal histogram to the horizontal projection of the text line such that the square error between the ideal and the real histogram is minimized. The parameters of the ideal histogram include the position of the upper and lower baseline of the writing. The bounding box of the whole line together with these two baselines define three disjoint areas (upper, middle, and lower text area). Each of these areas is scaled in vertical direction to a predefined size.

For horizontal scaling, the number of horizontal black/white transition are counted. This number can be compared to the average number determined over all text lines of the database, which is determined off-line. By computing the ratio between the average number and the number present in the actual text line, a scaling factor for the horizontal direction is obtained.

In Fig. 3 a fragment of a text line is shown before (above) and after normalization (below).

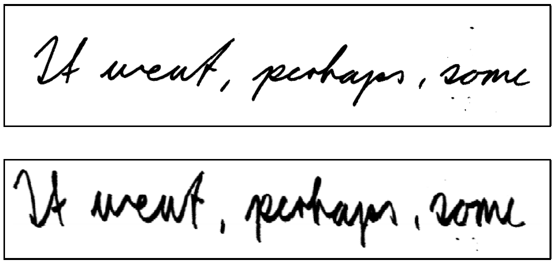

Figure 3. Original image of a text line (top); normalized text line (bottom).

3 Feature Extraction

For feature extraction, a sliding window technique is used. A window of one column width and the image's height is moved from left to right over each text line. At each position of the window nine geometrical characteristics of the image in the window are determined.

The first three features are the number of set pixels in the window, the center of gravity and the second order momentum. This set describes how many pixels in which region of the window are, and how they are distributed. They characterize the window from a global point of view.

For a more detailed description of the window, features four to nine give the upper and the lower contour in the window, the orientation of the contour, the number of black-white transitions in vertical direction and the number of black pixels between the upper and lower contour. Notice that all these features can be easily computed from the image of a text line.

4 Recognition

The recognizer used in this work is based on hidden Markov models (HMM) [10]. Because we deal with large vocabularies in this work, it is not possible to build and train an HMM for each word. Consequently, character models are used, which allow to share training data.

To achieve optimal recognition results, the character HMMs have to be fitted to the problem. In particular the number of states, the possible transitions and the output probability distributions have to be chosen. Based on empirical studies it was decided to use 14 states per character model, organized in a linear fashion with continous output distributions. For training the Baum-Welch algorithm [10], applied on whole text lines, is used.

In the recognition phase the character models are concatenated to words according to the underlying vocabulary, and the words are connected to sentences (see Sec. 4.1). Thus a recognition network is obtained, in which the best path can be found with the Viterbi algorithm [10]. From the optimal path in the network the desired recognition result, i.e. the most likely sequence of words from the vocabulary, can be extracted.

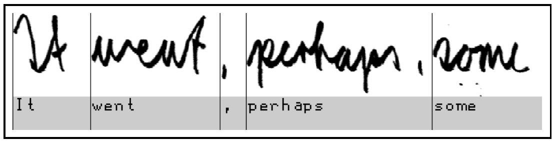

Figure 4. Text image with recognition results and word boundaries.

One crucial feature of the system described in this paper is that text lines are not segmented into single words during preprocessing, but the segmentation of the line into words is delivered by the HMM as a byproduct of the recognition process (see Fig. 4). Thus the difficult problem of segmentation without any prior knowledge can be avoided.

4.1 Statistical Language Model

In natural language, the frequency of words is not equally distributed. Neither are position and sequence of words random. Therefore, additional knowledge about the language can be introduced in the recognition of handwritten sentences. In this paper three language models are studied: a *simple sentence*, a *word unigram* and a *word bigram* model. In the simple sentence model each word has the same probability and it can occur at any position. Hence the probability $p(w_i) = 1/n$ of each word w_i depends only on the size n of the used vocabulary. To compute unigram and bigram probabilities, samples of training data are needed. For this task the Lancaster-Oslo/Bergen (LOB) corpus [11] is used, which contains about 1'000'000 word instances[1]. From this corpus the following three quantities are obtained: N - the total number of word instances of the considered vocabulary; N_i - the number of word instances of word w_i of the vocabulary; $N_{i,j}$ - the number of instances of word pair (w_i, w_j) occurring in the corpus.

The word unigram probability $p(w_i)$ and the bigram probability $p(w_j|w_i)$ that word w_j occurs if word w_i has been observed, can be computed in the following way:

$$p(w_i) = \frac{N_i}{N} \tag{1}$$

$$p(w_j|w_i) = \frac{N_{i,j}}{N_i}. \tag{2}$$

During Viterbi decoding, these probabilities weight the words in the vocabulary. The unigram probability $p(w_i)$

[1] Note that all handwritten training and test data is also based on this corpus [9].

is applied to a word at the beginning of a text line where no contextual knowledge is available. From the second word on bigram probabilities $p(w_j|w_i)$ are applied.

Although the number of word instances in the LOB corpus seems quite large at the first glance, it is not sufficient to estimate all probabilities - particularly the probabilities of the bigram model - reliably. Therefore, a smoothing technique has been applied in the process of computing $p(w_i)$ and $p(w_j|w_i)$. Details of this technique have been described in [12, 13].

4.2 Perplexity

One way to compare different language models with each other is by means of recognition experiments. However, such experiments can be computationally quite costly. Another way is to compare language models analytically based on perplexity, $\mathbf{P}$, which is defined as follows [14]:

$$\mathbf{P} = 2^{\mathbf{H}} = p(s)^{-1/l}, \quad \text{where} \quad \mathbf{H} = -\lim_{l \to \infty} \frac{1}{l} \log_2 p(s) \tag{3}$$

In these formulas s is a sentence of length l of the considered language. The probability for a sentence using the bigram sentence model is then computed as follows:

$$p(s) = p(w_1) \prod_{i=2}^{l} p(w_i|w_{i-1}) \tag{4}$$

with $p(w_i)$ and $p(w_i|w_{i-1})$ being the word and the word pair occurrence probability, respectively. Intuitively, $\mathbf{P}$ can be interpreted as the average number of possible successors of a word. Clearly, the smaller the perplexity is, the better is a language model.

One shortcoming of the perplexity as a measure of the quality of language models is its dependence on the vocabulary size. Obviously, the larger the vocabulary is, the larger is the average number of possible successors of a word expected to be. To overcome this dependency, one can use the normalized perplexity, $\bar{\mathbf{P}}$, which is obtained by dividing $\mathbf{P}$ by the vocabulary size n:

$$\bar{\mathbf{P}} = \frac{\mathbf{P}}{n}, \tag{5}$$

Notice that for each language model $\bar{\mathbf{P}} \in [0, 1]$.

5 Experiments and Results

In our experiments we particularly wanted to investigate the influence of the language model on the recognition task with an increasing vocabulary size. Therefore language models for different vocabulary sizes were generated. For a reliable training of the character models, the data set used for training and testing was increased corresponding to the vocabulary size (see Tab. 1). Each data set, corresponding to one row in Table 1, was split into 5 sets of equal size, four of which were used for training and the fifth for testing. By cyclic cross validation, each set was used once for testing.

Vocabulary size	Number of Handwritten Pages	Number of Textlines	Number of Word Instances	Number of Writers
412	59	541	4523	6
2346	116	991	8789	ca. 50
2703	136	1200	11000	ca. 70
3411	171	1517	14806	ca. 80
4409	280	2557	21462	ca. 140
7719	574	4333	44019	ca. 250

Table 1. Vocabulary size and the related system configurations.

For each vocabulary size, three language models were created: the simple sentence model, the unigram model and the bigram model. Because the simple sentence model includes no linguistic information at all, it was used as a basis for comparison.

In the first experiment, the perplexity of each language model was computed, using the LOB corpus (see Tab. 2). For the simple sentence model perplexity is, by definition, identical to the vocabulary size (see column 1). The values obtained for the unigram and bigram models are given in columns 2 and 3 of Table 2. As expected, with growing vocabulary size, the perplexity increases for all three language models. The normalized perplexity of the simple senctence model is always equal to one, regardless of the vocabulary size. For the unigram and bigram models it is shown in columns 4 and 5 in Table 2. In contrast to columns 2 and 3, the normalized perplexity decreases with an increasing vocabulary size. This mean that the relative number of words, which can be chosen under a language model with a large vocabulary, is smaller than for a small vocabulary. So the probability to choose the right word is higher. As a consequence, the influence of the language model on the recognition performance is stronger for large vocabularies than for small ones. The values in Table 2 are visualized on a logarithmic scale in Figure 5.

To verify the usefulness of the perplexity and the normalized perplexity as measures for language models, recognition experiments were done. For each vocabulary size all three language models were tested on the IAM-database [9]. In Table 3 the recognition rates are listed.

For the system configurations using the simple sentence model it can be observed that the word recognition rate decreases for increasing vocabulary size. For a system using 412 words, 78.53% of the words were recognized correctly,

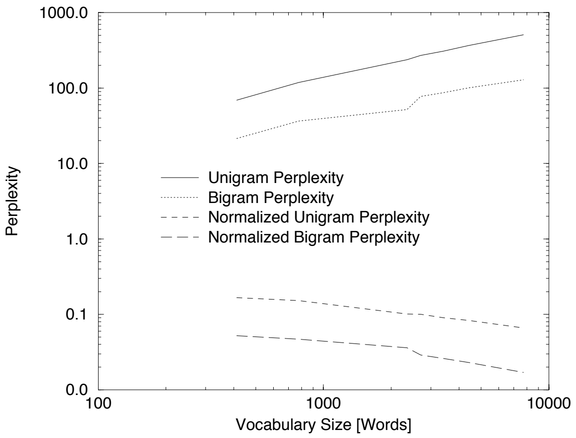

Figure 5. Normalized perplexity for different language models and vocabulary sizes.

Vocabulary	Perplexity		Normalized Perplexity	
size	Unigram model	Bigram model	Unigram model	Bigram model
412	68.8	21.5	0.167	0.052
2346	237.2	51.7	0.101	0.036
2703	269.7	77.3	0.100	0.029
3411	308.3	87.0	0.090	0.026
4409	365.9	100.3	0.083	0.023
7719	508.7	128.6	0.066	0.017

Table 2. Normalized perplexity for different language models and vocabulary sizes.

while for a 7719 word vocabulary the recognition rate drops down to 40.47%.

Using unigram models, only small improvements were measured compared to the result obtained with the simple sentence model. For a system with 412 words the recognition rate was almost equal to the one obtained with the simple sentence model, while for 7719 words an improvement of 1.66% was observed.

In the fourth column of Table 3, the recognition rates under the bigram model are listed. For all vocabulary sizes significant improvements can be measured using the additional information provided by word pair occurrence probabilities. For the 412 word vocabulary the word recognition rate rises from 78.53% to 81.27%. Using a system with 7719 words, the influence of the language model is much stronger. In this configuration 60.05% of the words are recognized correctly when bigrams are used, compared to 40.47% in case of the simple sentence model. The fifth column (Δ) in Table 3 gives the difference between the bigram model and the simple sentence model.

Vocabulary	Word Recognition Rate[%]			Δ
size	Simple sentence model	Unigram-model	Bigram-model	
412	78.53%	78.57%	81.27%	2.74%
2346	55.01%	57.76%	65.30%	10.29%
2703	51.44%	52.15%	63.94%	12.44%
3411	50.72%	51.71%	63.39%	12.67%
4409	47.58%	48.63%	61.68%	14.10%
7719	40.47%	42.13%	60.05%	19.58%

Table 3. Recognition rate for different language models and vocabulary sizes according to Tab. 2.

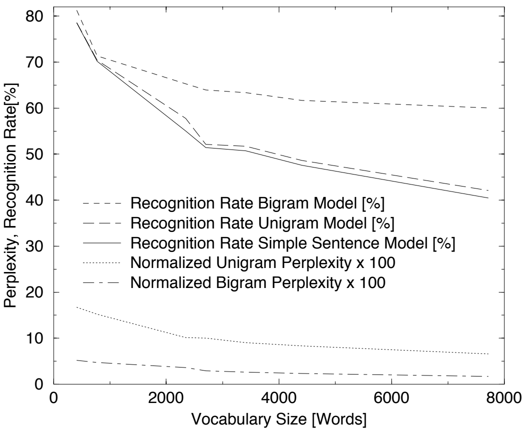

Figure 6. Word Recognition rate and perplexity for different language models and vocabulary sizes.

The relation between the recognition behavior and the normalized perplexity over the vocabulary size is illustrated in Figure 6. The behavior predicted by the theorectical analysis using the normalized perplexity becomes clearly visible in this graph. The normalized perplexity becomes smaller as the vocabulary grows. At the same time, the increase in recognition performance ganied through the language model becomes more pronounced, particularly for the bigram model.

6 Conclusions

In this paper we have presented a recognition system for unconstrained handwritten text. In contrast with other systems the segmentation of the text lines into individual words is not done in the preprocessing. The word boundaries are obtained as a byproduct of the recognition process, which is based on hidden Markov models. In our work a hidden Markov model is constructed for each character. These models are concatenated to word models according to the underlying vocabulary, and to word sequences using a language model which provides additional information.

A special feature of the system described in this paper is the integration of a statisitcal language model into the HMM framework. The perplexity and its normalized form have been used to compare different language models to each other. It has been observed that the normalized perplexity decreases for an increasing vocabulary size. This means that for larger vocabularies fewer words are possible to be chosen as the successors of a given word. From this observation the conclusion can be drawn that language models have a stronger influence on the recognition process if larger vocabularies are involved.

This behavior of language models that was theoretically predicted could be empirically confirmed by experiments. For a small size vocabulary an improvement of 2.74% from 78.53% to 81.27% was obtained through the use of a bigram language model, while for a vocabulary containing 7719 words, the recognition rate increased from 40.47% to 60.05%.

It can be conjectured that handwriting recognition tasks that involve larger vocabularies will play a more dominant role in the near future. Hence the importance of language models will increase. A challenging problem left to future research is the extension of the language models used in this paper to n-grams where $n > 2$.

References

[1] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai, and L. Lam. Computer recognition of unconstrained handwritten numerals. *Special Issue of Proc. of the IEEE*, 7(80):1162–1180, 1992.

[2] J.-C. Simon. Off-line cursive word recognition. *Special Issue of Proc. of the IEEE*, 80(7):1150–1161, July 1992.

[3] N. Gorski, V. Anisimov, E. Augustin, D. Price, and J.-C. Simon. A2ia check reader: A family of bank check recognition systems. In *5th Int. Conference on Document Analysis and Recognition 99, Bangalore, India*, pages 523–526, 1999.

[4] G. Kaufmann and H. Bunke. Automated reading of cheque amounts. *Pattern Analysis and Application*, 3(2):132–141, 2000.

[5] A. Kaltenmeier, T. Caesar, J.M. Gloger, and E. Mandler. Sophisticated topology of hidden Markov models for cursive script recognition. In *Proc. of the 2nd Int. Conf. on Document Analysis and Recognition, Tsukuba Science City, Japan*, pages 139–142, 1993.

[6] B. Lazzerini, F. Marcelloni, and L.M. Reyneri. Beatrix: A self-learning system for off-line recognition of handwritten texts. *Pattern Recognition Letters*, 18(6):583–594, June 1997.

[7] G. Kim, V. Govindaraju, and S.N. Srihari. Architecture for handwritten text recognition systems. In S.-W. Lee, editor, *Advances in Handwriting Recognition*, pages 163–172. World Scientific Publ. Co., 1999.

[8] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1):41–52, January 1994.

[9] U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *Proc. of the 5th Int. Conf. on Document Analysis and Recognition, Bangalore, India*, pages 705–708, 1999.

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

[11] S. Johansson, G.N. 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.

[12] U.-V. Marti and H. Bunke. Unconstrained handwriting recognition: Language models, perplexity, and system performance. In *Proc. of the 7th Int. Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands*, pages 463–468, 2000.

[13] U. Marti. *Offline Erkennung Handgeschriebener Texte*. PhD thesis, University of Bern, Switzerland, 2000.

[14] F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann Publishers, Inc., 1990.