# AN INVESTIGATION OF THE USE OF TRIGRAPHS FOR LARGE VOCABULARY CURSIVE HANDWRITING RECOGNITION

*Andreas Kosmala, Jörg Rottland, Gerhard Rigoll*

Gerhard-Mercator-University Duisburg, Germany
Faculty of Electrical Engineering, Dept. of Computer Science
E-mail: {kosmala,rottland,rigoll}@fb9-ti.uni-duisburg.de
http://www.fb9-ti.uni-duisburg.de

## ABSTRACT

This paper presents an extensive investigation of the use of trigraphs for on-line cursive handwriting recognition based on Hidden Markov Models (HMMs). Trigraphs are context dependent HMMs representing a single written character in its left and right context, similar to triphones in speech recognition. Looking at the great success of triphones in continuous speech recognition ([1]-[3]), it was always a challenging and open question, if the introduction of trigraphs could lead to substantially improved handwriting recognition systems. The results of this investigation are indeed extremely encouraging: The introduction of suitable trigraphs led to a 50% relative error reduction for a writer dependent 1000 word handwriting recognition system, and to a 35% relative error reduction for the same system with an extended 30000 word vocabulary for cursive handwriting recognition.

## 1. INTRODUCTION

Regarding unconstrained, cursive written words it seems obvious, that the context of a character within a word has a basic influence on the character itself. Fig. 1 shows some examples how a handwritten character can occur in different styles depending on its left and right context.

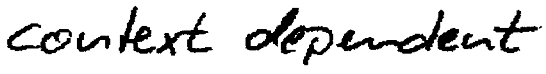

Fig. 1: Example for the context dependency of characters in unconstrained cursive handwriting.

The grapheme /n/ for instance, shows completely different characteristics in the context of /o/ and /t/ in the word "context" than in the context of /e/ and /d/ in the word "dependent". Similar effects can be observed for the grapheme /t/, which differs in the context of /n/ and /e/ in the word "context" from the /t/'s at the word ends. This example gives an idea of the potential of context dependent modeling to increase the accuracy in handwriting recognition systems, which is directly emphasized and discussed in this paper.

In the following section we give a brief description of the handwriting recognition system and the database which is used for training and test. The third section describes the introduction of trigraphs and the problems, which arise with this approach. Some strategies are discussed to solve these specific problems, and results are presented before we give a summarizing conclusion in the last section.

## 2. SYSTEM OUTLINE

Our handwriting system is based on discrete HMMs and has several special features in order to obtain high performance cursive handwriting for large vocabulary recognition tasks [4].

### 2.1 Feature Extraction

After the handwriting was captured by a digitizer tablet with a constant sample rate, the trace of the pen is given as a sequence of Cartesian coordinates. To suppress the influence of the writing speed, this trajectory is spatially re-sampled with vectors of constant length [4]. This re-sampled sequence of vectors with constant length and different orientations, is the base for the extraction of three on-line features and one off-line feature.

Because absolute position of the vector is irrelevant, one single vector is completely described by its orientation. The first on-line feature vector, which is extracted from the re-sampled trajectory consists of the sine and cosine of the angle of the current re-sampling vector. The second feature vector includes the sine and cosine of the differential angle between two re-sample points. The third on-line feature is a binary feature and describes the pen pressure, which can be zero (negative pressure=pen lifted) or one (positive pressure=pen set down).

The off-line feature is a bitmap of 30x30 pixels. The bitmap is centered around the current re-sampling position and is slided jointly along the pen trajectory during re-sampling. A spatial sub-sampling of the bitmap results in a window of 3x3 blocks, which is used as a nine dimensional off-line feature vector. The additional bitmap has been recently introduced and already improved the original system presented in [4]. Therefore, in the following all other comparisons are related to the monograph based system using this additional off-line feature. In this configuration, we obtained a recognition rate of 96.76% for the 1000 word vocabulary task and 92.47% recognition rate with a 30000 word vocabulary. The use of the off-line feature resulted especially in combination with trigraphs in an improved recognition: The trigraph based system obtained 96.24% recognition rate for the 1000 word recognition task *without* off-line feature, while the recognition rate of the same system *with* additional bitmap was 98.39%. This means a relative error reduction of 57% with the bitmap approach.

Multiple codebook technique is used for quantizing the angle, the differential angle and the sub-sampled bitmap of 3x3 blocks resulting in three different discrete feature streams plus the pressure feature. As vector quantizer the well known k-means algorithm is used with different codebook sizes for each feature vector.

### 2.2 Modeling

Each of 80 baseline characters is represented by a single linear discrete HMM consisting of 12 states. This unusual high number is necessary in order to cope with the non stationary data resulting from spatial re-sampling of the pen input.

### 2.3 Database

The database, which is written by a male writer consists of 125 sentences including 2000 words for training and is represented by 500000 feature vectors. For testing, 200 written words from another corpus are used. The active vocabulary is either 1000 words for a more compact and quicker performance test case, or 30000 words for a very complex large vocabulary handwriting recognition task.

## 3. TRIGRAPH MODELING

The goal was now to further improve this already powerful handwriting recognition system by introducing trigraphs as basic units for representing the characters in the system.

### 3.1 Clustering

The usual way for building trigraphs in such a system would be as follows:

- Trigraph generation from the trained monographs.
- Re-estimation of the generated trigraphs.
- Parameter-clustering.
- Re-estimation of the clustered trigraphs.

After Baum-Welch re-estimation of the monographs a set of context dependent trigraphs is generated in the first step. Depending on the used vocabulary the monograph models are copied to corresponding trigraph models. This resulted in a set of 2300 (8900) trigraph models for the 1000 (30000) word vocabulary. After re-estimation in step two, the third step would be a data driven state clustering. Data-driven clustering means, that either those states share the same parameter set which have a small distance in their parameters or which are outlying in the parameter space. Outlying states were shared with the nearest cluster in the parameter space. Building trigraphs leads to a very large number of HMMs, whereby the number of feature vectors per HMM drops drastically. Clustered HMMs share the same parameters in order to increase the number of feature vectors which are available for training relative to the number of free system parameters. Arguing that the context has a small influence to the center states of a trigraph, clustering can be applied to all center states across all trigraphs with the same central grapheme. In this way, a suitable clustering could avoid inaccurate parameter estimation in step four.

With this procedure a maximum recognition rate of 90.32% could be obtained for the 1000 word vocabulary (see Tab. 1, third column). This means a disappointing relative error reduction of -198.8%.

### 3.2 Selected Trigraphs

The decreased recognition rate with the clustered trigraphs compared to the recognition rate of the baseline system with monographs results from two main problems, which arise with the introduction of trigraphs. As mentioned above, there is a significantly enlarged number of models. This leads to the effect, that most models were just represented by one or two examples in the training set. Fig. 2 shows, that in case of the 1000 word vocabulary for 850 trigraph models only one example appears in the training set ($N$=1) and that 310 trigraph models will be trained with only two examples ($N$=2).

|  | baseline | clustered | selected trigraphs |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| $N_{min}$ | - | - | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
| add. models | 0 | 2220 | 240 | 133 | 86 | 58 | 40 | 33 | 28 |
| accuracy | **96.76** | 90.32 | 94.62 | 95.16 | 97.31 | **98.39** | 97.85 | 97.85 | 97.85 |
| rel. error red. | 0 | -198.8 | -66.1 | -49.4 | 17.0 | **50.3** | 33.6 | 33.6 | 33.6 |

Tab. 1: Recognition rate and error reduction of trigraph-models (1000 word vocabulary).

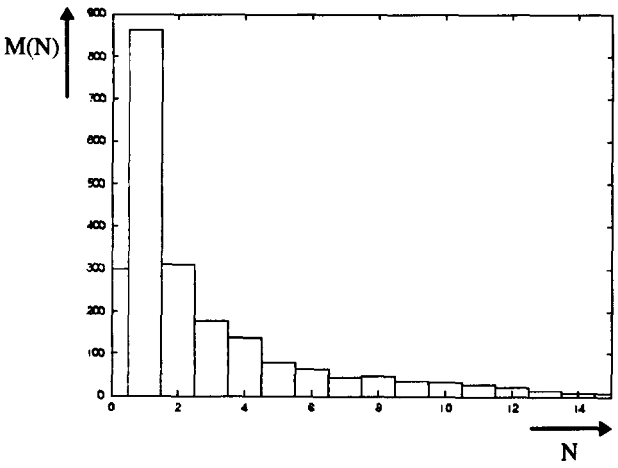

Fig. 2: Frequency M of trigraph models with N examples in the training set (1000 word vocabulary).

At this point of research the used training database of 2000 words seemed too small for building a powerful trigraph system. Also can be found in [5], that a training set between 1000 and 2000 words is just enough for a monograph system.

We were expecting that we could easily solve this sparse data problem by reducing the number of free HMM parameters through appropriate clustering methods, as mentioned before. To our surprise, even after successful clustering we were left with the disappointing drop of the recognition rate from 96.76% to 90.32% as documented in columns 2 and 3 of Table 1. Through a more detailed analysis of Fig. 2, we detected later the second major problem of our trigraph approach for handwriting recognition: The leftmost part of the function in Fig. 2 shows that for $N$=0, the number of M is equal to 300. This can be interpreted in the following way: In the 1000 words used for our recognition task, there are 300 trigraphs for which we have no training examples in our training database. This problem of "unseen trigraphs" typically occurs in open vocabulary tasks, where the vocabulary used in the training and test data differs substantially. In the 30000 word vocabulary, this problem is even much more evident, because in this case we obtained 6900 unseen trigraphs compared to 2000 seen trigraphs from the training set. The major problem results now from the fact, that during clustering many unseen trigraphs are clustered together with seen trigraphs and are then trained with the data assigned to the seen trigraphs.

For instance, if an unseen trigraph */end/* in the word "dependent" in Fig. 1 is clustered together with a seen trigraph */ont/* in the word "context" (where the central grapheme */n/* is equal for both trigraphs), the trigraph */end/* will be later trained with training examples for */n/* in the left context with */o/* and the right context with */t/*, which will be completely senseless for */end/*.

There are two solutions to that problem: One solution is to hold on to the clustering procedure and to make sure that in our example, the unseen trigraph */end/* receives the parameters of the monograph */n/*. The other solution is to omit the clustering procedure and to look for other ways in order to keep the number of free HMM parameters small. We have chosen this second way, because it is easier to implement in our system, but we will investigate the first way in a later stage.

Our approach has been the following: Instead of introducing all possible trigraphs, we introduce only a reduced set of selected trigraph models. This set contains only trigraphs, which are represented at least $N$ times in the training database, where $N$ has to be greater than a specified number $N_{min}$. For $N_{min}$ = 15 for instance, 133 trigraphs can be found, which have more than 15 examples in the training set (see Tab. 1). Trigraphs that would have less than 15 examples were left as monographs. Tab. 1 shows the recognition rate and the relative error reduction of the system with selected trigraphs depending on $N_{min}$. It can be seen, that a maximum relative error reduction of more than 50% with a recognition rate of 98.39% for the 1000 word vocabulary is obtained for $N_{min}$ = 25, which delivers 58 additional trigraph models. Just so the test with the challenging 30000 word vocabulary should encourage one to apply trigraphs in on-line character recognition (OLCR) systems: the recognition rate was increased from 92.47% up to 95.16% with a relative error reduction of 35.7% in this case.

For verification, we recorded a similar database with the same text for training and test with a second (female) writer. We obtained for the 1000 word recognition task with monographs a recognition rate of 90.32%. With the use of selected trigraphs and *Nmin* = 25, we could increase the recognition rate up to 93.58%, which is equivalent to a relative error reduction of 34.0%. It should be noted, that the writer dependent system was optimized for the first writer and absolute recognition rates are correspondingly higher. Nevertheless we could even reach an appropriate error reduction for the second writer and could demonstrate the reliability of this approach.

## 4. CONCLUSION

From these clear results two important conclusions can be derived. First, that similar to speech recognition the introduction of context dependent models promise an enormous potential for improving OLCR-systems. Second, that a skillful use of trigraphs enables even on relative small databases amazing gains. Similar results are anticipated for the writer independent task. Regarding the variability of handwriting between different writers, the context of a character could deliver a valuable additional information for recognition and gains should be expected to be even higher as in the writer dependent case. Further investigations of context dependent models would consider also the use of bigraphs. The combination of monographs, left- and right-context bigraphs and trigraphs seems also very promising. The selection which kind of model is used, could be based on the number of available examples in the training set, as described above. Bigraphs would form an intermediate stage between the large number of sparsely represented trigraphs and the small number of frequently represented monographs.
The introduction of trigraphs offers a large potential for future improvements of OLCR-systems. Trigraphs with state-clustering could be one important topic on the way to robust recognition systems, provided that the problem with unseen trigraphs will be solved. This could either be done by the use of closed vocabularies or by the use of non-data-driven clustering algorithms like tree-based clustering, for instance. The first solution implies the disadvantage, that the training set must be updated with the vocabulary, so that the second solution seems to offer more long term flexibility.

In summary, it should be pointed out, that we believe that our investigation and the results in this paper is one of the first systematic investigations of the use of trigraphs for handwriting recognition, with two important outcomes: First, the introduction of trigraphs is not easy and has to be carried out very skillfully. Secondly - and what is more important: Context dependent HMMs may represent one of the most important potentials for future handwriting recognition systems. We will try to make full use of our discoveries and we hope to be able to present soon a handwriting system exploiting the full potential offered through context dependent character modeling techniques.

## REFERENCES

[1] L. Bahl, R. Bakis, P. Cohen, A. Cole, F. Jelinek, B. Lewis, R. Mercer: „Further Results on the Recognition of a Continuously Read Natural Corpus“, *Proc. IEEE-ICASSP, Denver, 1980, pp. 872-875.*

[2] R. Schwartz, J. Klovstadt, L. Makhoul, J. Sorensen: „Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition“, *Proc. IEEE-ICASSP, San Diego, 1984, pp. 35.6.1-35.6.4.*

[3] K. Lee: „Context Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition“, *Trans. on Acoustics, Speech, and Signal Processing, Vol. 38, No. 4, 1990, pp. 599-609.*

[4] G. Rigoll, A. Kosmala, J. Rottland, Ch. Neukirchen: „A Comparison between Continuous and Discrete Density Hidden Markov Models for Cursive Handwriting Recognition“, *Proc. IEEE-ICPR, Vienna, 1996, Vol. 6, pp. 205-209.*

[5] J. Subrahmonia, K. Nathan, M. P. Perrone: „Writer Dependent Recognition of On-Line Unconstrained Handwriting“, *Proc. IEEE-ICASSP, Atlanta, 1996, pp. 3478-3481.*