Large Vocabulary Off-Line Handwritten Word Recognition
[ Reconnaissance Hors–Ligne de Mots Manuscrits Dans un Lexique de Très Grand Dimension ]

Alessandro L. Koerich
[ École de Technologie Supérieure ]

The Early Beginning

In 1998, the baseline recognition system developed by A. El–Yacoubi at the SRTP had the following performance:

- 100–word vocabulary
  - Accuracy: 95.89% of the words are correctly recognized (4,481 out of 4,674)
  - Speed: 2 sec/word

- 30,000–word vocabulary
  - Accuracy: 73.70% of the words are correctly recognized (3,445 out of 4,674)
  - Speed: 8.2 min/word
  - 26 days to recognize the whole test set

The Story and The Goal

In this presentation I will show you how we have addressed the problems related to the accuracy and speed to build an off–line handwriting recognition system which has the following characteristics:
- Omniwriter
- Very–large vocabulary (80,000 words)
- Unconstrained handwriting
- Good recognition accuracy
- Good recognition speed

Presentation Outline

- Introduction
- Problem Statement
- Thesis Contributions
  - Word Recognition System Based on HMMs
    - Speed: Fast Two–Level HMM Decoding
  - Verification Module Based on NNs
    - Accuracy: Post–Processing Word Hypotheses
- Conclusion
Motivation

Current research has focused on relatively simple problems such as:

- Isolated characters, digits and digit strings
- Words in small and medium vocabularies
- Well-determined application domains
- Emphasis on the recognition accuracy

Limitations

We can conclude that large vocabulary handwriting recognition is a challenge research subject.

Recent Results in Handwriting Recognition

<table>
<thead>
<tr>
<th>Author</th>
<th>Method</th>
<th>Lexicon Size</th>
<th>Accuracy (%)</th>
<th>Test Set (#)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marti et al.  [30]</td>
<td>HMM</td>
<td>7,719</td>
<td>60.05</td>
<td>44,019</td>
<td>UNC, OMNI, 250 Writers</td>
</tr>
<tr>
<td>Cho et al. [7]</td>
<td>HMM</td>
<td>10,000</td>
<td>67.09</td>
<td>700</td>
<td>CUR, OMNI</td>
</tr>
<tr>
<td>Wimmer et al. [40]</td>
<td>NN/HMM</td>
<td>20,200</td>
<td>73.13</td>
<td>1,500</td>
<td>CUR, WD</td>
</tr>
<tr>
<td>Brakensiek et al. [3]</td>
<td>HMM</td>
<td>30,000</td>
<td>89.2</td>
<td>800</td>
<td>CUR, WD, 9 writers</td>
</tr>
<tr>
<td>Dzuba et al. [10]</td>
<td>DP</td>
<td>40,000</td>
<td>60.7</td>
<td>3,000</td>
<td>CUR, OMNI, 4 Writers</td>
</tr>
</tbody>
</table>


Applications of LVHR

- Postal applications (addresses)
- Reading of handwritten notes
- Fax transcription
- Generic text recognition
- Information retrieval
- Reading of handwritten fields in forms
- Pen–pad devices

We can conclude that large vocabulary handwriting recognition is a challenge research subject.
Presentation Outline

- Introduction
- **Problem Statement**
- Thesis Contributions
  - Word Recognition System Based on HMMs
    - Speed: Fast Two-Level HMM Decoding
  - Verification Module Based on NNs
    - Accuracy: Post-Processing Word Hypotheses
- **Conclusion**

Why Handwriting Recognition is Difficult?

Why Handwriting Recognition is Difficult?

The complexity of the recognition process in the case of unconstrained handwriting is:

\[ C = O(TN^2HV) \]

Assuming typical values of the parameters

\(T=30, L=10, N=10, V=80,000, H=2\)

and a standard Viterbi algorithm

**But current personal computers can perform between 1 GFLOPS and 3 GFLOPS!**

Current Methods for LVHR [speed]

- Lexicon pruning (prior to the recognition)
  - Application environment
  - Word length and shape
- Organization of the search space
  - Lexical tree x Flat lexicon
- Search strategy
  - Viterbi beam search
  - A*
  - Multi-pass

Most of these methods are not very efficient or/and they introduce errors which affect the recognition accuracy.
Current Methods for LVHR [accuracy]

- Improvements in accuracy are associated with:
  - Feature set
  - Modeling of reference patterns
  - More than one model for each character class
  - Combination of different feature sets / classifiers

The complexity of the recognition process has been steadily increasing with the recognition accuracy.

The Challenge

- We have to account for two aspects that are in mutual conflict: recognition speed and recognition accuracy
- In order to build an omniwriter large vocabulary off-line handwritten word recognition system (80,000 words)

It is relatively easy to improve the recognition speed while trading away some accuracy. But it is much harder to improve the recognition speed while preserving (or even improving) the original accuracy.

Methodology: Speed

- Build a LV word recognition system based on HMMs to generate a list of $N$-best word hypotheses as well as the segmentation of such word hypotheses into characters
- **Problem:** Current decoding algorithms are not efficient to deal with large vocabularies
- **Solution:** Speedup the recognition process using a novel decoding strategy that reduces the repeated computation and preserves the recognition accuracy

Methodology: Accuracy

- Build a verification module to post-process the $N$-best word hypotheses generated by the LVS which uses a different recognition strategy that focuses on the weaknesses of the HMM approach.
- **Problem:** The verification module should not cause delays in the recognition process
- **Solution:** Model segmented characters with NNs and rescore the $N$-best word hypotheses based on the combination of character probabilities
Methodology: Accuracy

- Combine the LV word recognition system based on HMMs with the verification module based on NNs to improve the overall recognition accuracy.
- Reject a word hypothesis based on the composite score provided by the recognition–verification system to further improve the recognition accuracy.

Presentation Outline

- Introduction
- Problem Statement
- Thesis Contributions
  - Word Recognition System Based on HMMs
    - Speed: Fast Two–Level HMM Decoding
  - Verification Module Based on NNs
    - Accuracy: Post–Processing Word Hypotheses
- Conclusion

Thesis Contributions

Word Recognition System Based on HMMs

- Use some modules of a baseline recognition system
- Segmentation–recognition approach
- Lexicon–driven approach where character HMMs are concatenated to build up words according to the lexicon
- Fast two–level HMM decoding algorithm to generate a list with the $N$–best word hypotheses as well as their definitive segmentation into characters
- Global recognition approach to account for unconstrained handwriting
Pre-Processing

- Original image
- Baseline slant normalization
- Character skew correction
- Lowercase character area normalization
- Final image after smoothing

Segmentation & Feature Extraction

- Loose segmentation
- Global features
- Contour transition histogram features
- Segmentation features

Character Model and Training

- Characters are modeled by 10-state HMMs
- Observations are emitted along transitions
- Baum-Welch algorithm to estimate the best parameter values of the character HMMs

Other Contributions

- Multiple Character Models
- Lexicon-Driven Level Building Algorithm
- Constrained Level Building Algorithm

These contributions are detailed in:
- Ph.D. Thesis
- Proc. of the ICAPR 2001 [Koerich01]
- IJDAR [Koerich02]
Fast Two–Level HMM Decoding (1)

We have observed that there is a great number of repeated computation during the decoding of words in the lexicon.

The current algorithms decode an observation sequence in a time–synchronous fashion; the probability scores of a character ($\lambda_i$) within a word depends on the probability scores of the immediate preceding character ($\lambda_{i-1}$);

During the recognition is it possible to decode the character “a” only once since it is always represented by the same character model?

Fast Two–Level HMM Decoding (2)

To solve this problem of repeated computation, we have proposed a new algorithm that breaks up the decoding of words into two levels:
- First Level: Character HMMs are decoded considering each possible entry and exit point in the trellis and the results are stored into arrays.
- Second Level: Words are decoded but reusing the results of first level. Only character boundaries are decoded.

FTL HMM Decoding: First Level (4)
We do not need to decode HMM states which is the most complex computation during the decoding (N^2T).

Performance of the Word Recognition System

- Two Lexicons: 36,015 words and 85,092 words
- Test set: 4,674 unconstrained words (city names)
- Three Platforms: SUN Ultra1, SUN Enterprise 6000, and AMD Athlon 1.1GHz

Word Recognition System Based on HMMs

- The output of the Word Recognition System Based on HMMs is a list with the N-best word hypotheses, the segmentation of such word hypotheses into characters and likelihoods.
Distributed Recognition Scheme

We explore the potential for speeding up the recognition process via distributed processing.

Split the lexicon, partitioning the recognition task among several processors.

The speedup in the recognition process is proportional to the number of processors.

**Thesis Contributions**

Performance of the Distributed Recognition Scheme

**Summary of Speed Improvements**

30,000–word vocabulary

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Computational Complexity</th>
<th>MFLOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Viterbi</td>
<td>$HTN$</td>
<td>4,800</td>
</tr>
<tr>
<td>FSTree</td>
<td>$T^2(N^2M + 800)$</td>
<td>500</td>
</tr>
<tr>
<td>DS FSTree</td>
<td>$T^2(N^2M + 500)$</td>
<td>50</td>
</tr>
</tbody>
</table>

- $V$ is the dominant variable.
- $T$ is quadratic but its magnitude is low.
- The FS is advantageous while $T < N^2$. 

30,000–word vocabulary
Presentation Outline

- Introduction
- Problem Statement
- **Thesis Contributions**
  - Word Recognition System Based on HMMs
    - Speed: Fast Two-Level HMM Decoding
  - Verification Module Based on NNs
    - Accuracy: Post-Processing Word Hypotheses
- Conclusion

Problems of the HMM Approach

- The features are not very discriminant at character level (trade–off between segmentation and recognition)
- Local information is somewhat overlooked
- Conditional independence prevents an HMM from taking full advantage of the correlation that exists among the observations of a single character

Thesis Contributions

**Solution ?**

- Power of the HMM approach to segment words into characters and an NN classifier to model isolated characters
- **Proposal**: To integrate HMMs (segmentation+recognition) and NNs (verification) to overcome the problems of particular methods.
- **Motivation**: 

Recognition–Verification Scheme
Motivation & Facts

Segmentation of words into characters which is carried out by the HMM recognizer is somewhat reliable – 8,005 out of 10,006 words images from the training dataset were correctly segmented (80%).

The difference in the recognition rate between the TOP 1 and TOP 10 word hypotheses is greater than 15% (80,000-word).

Neural Network Classifier

- MLP classifier (108–90–26)
- Uppercase + lowercase → Metaclasses
- Estimates Bayesian a posteriori probabilities given the character class

<table>
<thead>
<tr>
<th>Character Recognition Rate (%)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Upper</td>
<td>Lower</td>
<td>Mixed</td>
<td>Combination</td>
</tr>
<tr>
<td></td>
<td>Case</td>
<td>Case</td>
<td>Case</td>
<td>Upper+Lower</td>
</tr>
<tr>
<td>Training</td>
<td>97.26</td>
<td>95.01</td>
<td>96.71</td>
<td>96.27</td>
</tr>
<tr>
<td>Validation</td>
<td>92.09</td>
<td>88.88</td>
<td>90.08</td>
<td>89.23</td>
</tr>
<tr>
<td>Test</td>
<td>92.37</td>
<td>84.09</td>
<td>88.80</td>
<td>85.51</td>
</tr>
</tbody>
</table>

Probability Estimation by NN

- original image
- loose segmentation
- final segmentation
- probability estimation
Combination of Character Scores

\[ \Psi_{NN} = P(L_n \mid S_n) = \prod_{h=1}^{H} \frac{P(c_h \mid x_h)}{P(c_h)} P(L_n) \]

Combinations of HMM and NN Scores

Weighted Rule: \( P_{\text{COMB}} = \alpha \log(P_{\text{HMM}}) + \beta \log(P_{\text{NN}}) \)

\[ P_{\text{HMM}} = \frac{\Psi^n_{\text{HMM}}}{\sum_{n=1}^{N} \Psi^n_{\text{HMM}}} \]

\[ P_{\text{NN}} = \frac{\Psi^n_{\text{NN}}}{\sum_{n=1}^{N} \Psi^n_{\text{NN}}} \]

Reevaluation of the Word Recognition

<table>
<thead>
<tr>
<th>Lexicon Size</th>
<th>Word Recognition Rate (%)</th>
<th>Recognition Speed (sec/word)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOP 1</td>
<td>TOP 5</td>
</tr>
<tr>
<td>10</td>
<td>98.84</td>
<td>99.96</td>
</tr>
<tr>
<td>1k</td>
<td>91.01</td>
<td>96.32</td>
</tr>
<tr>
<td>10k</td>
<td>81.06</td>
<td>90.58</td>
</tr>
<tr>
<td>40k</td>
<td>73.23</td>
<td>84.64</td>
</tr>
</tbody>
</table>

PC AMD Athlon 1.1GHz 512MB RAM

85,092–word vocabulary (French + US + Italian + Brazilian + Québec city names)
Performance of the Recognition + Verification Approach

- Word Recognition (HMM)
- Verification (NN)
- Recognition + Verification (HMM + NN)

Word recognition rate based on:
- Output of the word recognition system alone (HMM)
- Output of the verification module alone (scores produced by the NN)
- Combination of the recognition and verification (HMM + NN)

Recognition + Verification

For a recognition task considering a 80,000-word vocabulary and TOP 10 word hypotheses:
- The word recognition system based on HMMs takes 16s to generate a list of the TOP 10 word hypotheses (Preprocessing + Segmentation + Feature Extraction + Recognition)
- The verification process takes less than 110ms (Feature Extraction + Recognition + Combination + Reranking)
- Therefore, the verification process takes less than 1% of the time required to generate the TOP 10 word hypothesis list.

Rejection

- We need only one word hypothesis at the end
- Is the word hypotheses provided by the recognizer trustworthy?
- We have used the probability scores of the TOP N word hypotheses to establish a rejection criteria

Rejection Rule

\[ P_2^* \leq P_1^* (1 + k) \]
Presentation Outline

- Introduction
- Problem Statement
- Thesis Contributions
  - Word Recognition System Based on HMMs
    - Speed: Fast Two-Level HMM Decoding
  - Verification Module Based on NNs
    - Accuracy: Post-Processing Word Hypotheses

Conclusion

We have built an omniwriter off-line handwritten word recognition system that deals efficiently with large and very-large vocabularies, unconstrained handwriting styles, and runs on personal computers with an acceptable performance.

BEFORE

- 30,000-word lexicon
  - Accuracy: 73.70%
  - Speed: 8.2 min/word

- 80,000-word lexicon
  - Accuracy: 68.65% (86%)
  - Speed: 16.2 min/word

NOW

- 30,000-word lexicon
  - Accuracy: 73.70%
  - Speed: 20.2 sec/word
    - 4.2 sec/word (Distr)

- 80,000-word lexicon
  - Accuracy: 78.05% (95%)
  - Speed: 63.2 sec/word

Summary of the Contributions (1)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Idea</th>
<th>Solution</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to improve the recognition speed while preserving the recognition accuracy</td>
<td>To avoid the repeated computation of the probabilities of the same character HMMs given an observation sequence</td>
<td>To breakup the computation of characters and words → Fast Two-Level HMM Decoding Algorithm</td>
<td>Speedup (24x to 120x) of the recognition process while maintaining exactly the same accuracy</td>
</tr>
</tbody>
</table>

Summary of the Contributions (2)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Idea</th>
<th>Solution</th>
<th>Results</th>
</tr>
</thead>
</table>
| How to improve the recognition accuracy without affecting the recognition speed achieved by the FS | To postprocess the N-best word hypothesis list by different recognition approach based on character recognition | Neural network classifier which estimates character probabilities and rescores the N-best word hypothesis list → Verification of Word Hypotheses | Improvement of 6.7% (10,000 words) to 9.4% (80,000 words) in the word recognition rate
Verification process# takes less than 1% of the time required to generate a TOP 10 word hypothesis list for an 80,000-word vocabulary*. |

*110msec on an PC Athlon 1.1GHz
*16sec on an PC Athlon 1.1GHz
Summary of the Contributions (3)

- Problem: How to find out if the word hypothesis ranked as TOP 1 is trustworthy (how to reduce the error rate)
- Idea: To reject a word hypothesis based on the composite score provided by the recognition-verification system
- Solution: Reject the word hypotheses based on the score of the TOP 1 and TOP 2 word hypothesis
- Results: Improvement of about 27% (80,000 words) in the word recognition rate by rejecting 30% of the word hypotheses

Conclusion

We have overcame the accuracy and speed problems to make LVHR feasible

- The Fast Two-Level HMM Decoding Algorithm has speeded up significantly the recognition process without affecting the accuracy
- The integration of two different classifiers (HMMs and NNs) has improved significantly the recognition accuracy without affecting the recognition speed

The proposed LV recognition system is one of the first recognition systems that deals with:

- Unconstrained handwriting
- Very-large vocabulary
- Ominiwriter

Future Work

- SPEED:
  - Use of heuristics into the fast two-level HMM decoding algorithm to further speedup the recognition process
  - Application of the FTL HMM decoding in other domains like speech recognition

- ACCURACY:
  - Improvement of the duration model of isolated characters
  - Verification of undersegmented and oversegmented characters
  - Automatic selection of the N-best word hypotheses

Conclusion and Implications

The results obtained are still far from meeting the throughput requirements of many applications

However, the results are very promising and they may stimulate further research on LV
Thank you for your attention!

**Multiple Character Models**

How to integrate multiple character models into the lexical tree search?

**The Recognition Process**

The use of the maximum approximation to select at each level only the more likely model.

- **Advantage**
  - Linear growth
  - \( H(L-L_s)V \)

- **Disadvantage**
  - Suboptimal
A Paradigm for HR

The Complexity of Handwriting Recognition

Assuming typical values of the parameters (T=30, L=10, M=10, V=80,000, H=2) and a standard Viterbi algorithm:

- **Generic Recognition Process**
  
  \[ C = O \left( T M^2 L V \right) = 2.4 \text{ GFLOPS} \]

- **With Two Writing Styles**
  
  \[ C = O \left( H T M^2 L V \right) = 4.8 \text{ GFLOPS} \]

- **Mixture of Writing Styles**
  
  \[ C = O \left( H^1 T M^2 V \right) = 245 \text{ GFLOPS} \]

Fast Two–Level Algorithm (1)

Fast Two–Level Algorithm (2)
Fast Two–Level Algorithm (3)

4. State Backtracking: for $L - 1, L - 2, \ldots, 1$:

$$
\hat{c}_k = c_k \phi^*(L) \\
\hat{c}_i = c_i \phi^*(i + 1) - 1 - \phi^*(i)
$$

(C.62)

Why Handwriting Recognition?

- To facilitate the transfer of information between people and machines
- Easy way of interacting with a computer
- To automate the processing of a great number of handwritten documents

Introduction

Requirements

- Focus on the weaknesses of the word recognition system (HMMs);
- Better discrimination of similar shapes;
- Results should be systematically integrable with the HMMs;
- The verification should not cause delays in the recognition process