Towards Unsupervised Learning for Handwriting Recognition

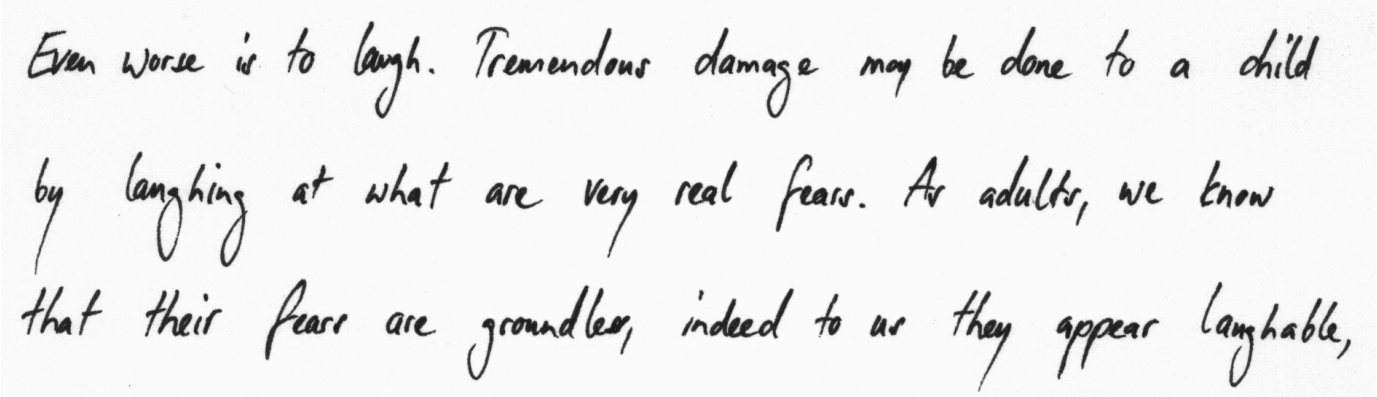
Michal Kozielski, Malte Nuhn, Patrick Doetsch, Hermann Ney

Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University
D-52056 Aachen, Germany

September, 2014

Introduction

The aim of **(modern) off-line handwriting recognition** is to obtain the transcription of an image containing a text.



- In the **supervised** scenario the perfect transcription to train the system is given.
- In the **unsupervised** scenario the transcription has to be uncovered by the system itself.

Outline

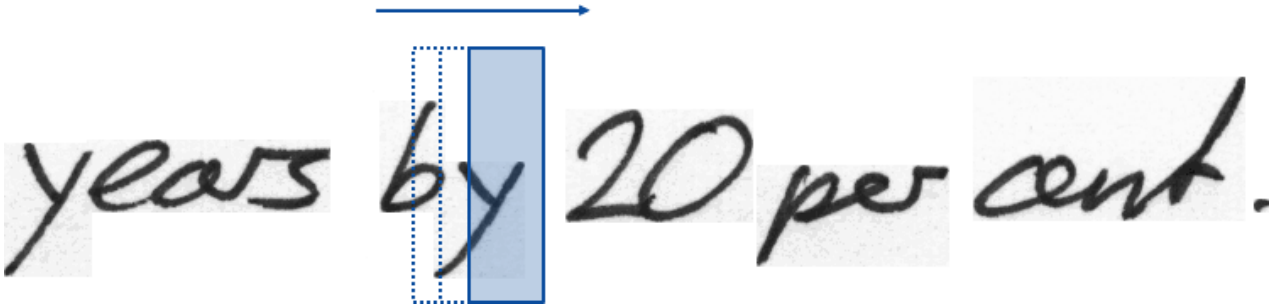
- Hidden Markov model training pipeline
- Unsupervised training
- Training approximations
- Experimental results

State-of-the-art

- Frinken '11 "Co-training for handwritten word recognition"
  - Semi-supervised learning, where an HMM and a BLSTM neural network try to improve each other.
- Knight '06 "Unsupervised analysis for decipherment problems"
  - Fully-unsupervised learning for machine translation using the EM algorithm.
- Kae '09 "Learning on the fly: Font-free approaches to difficult OCR problems"
  - Fully-unsupervised learning for machine-printed text using cipher-breaking algorithms.

System overview

The approach fits well into the standard HMM framework based on the **sliding window**.

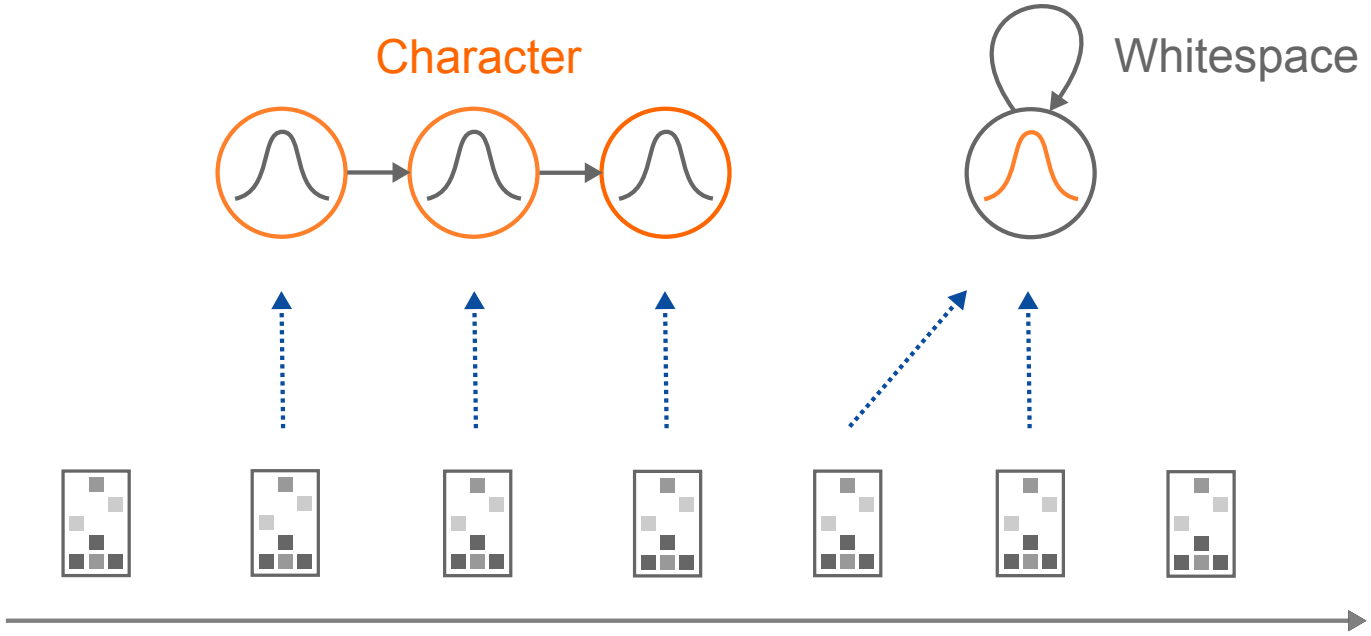


- A sequence of frames is extracted by moving an overlapping sliding window over a line of text.
- A feature vector consists of **gray-scale values** of all pixels in a frame (reduced by PCA to 20 components).

HMM revisited

Every character (one HMM model) encompasses multiple frames.

Recognition aims to find a sequence of models with the best score (best-first search).



Training method

Iteratively bootstrap model without any transcription in an expectation maximization fashion.

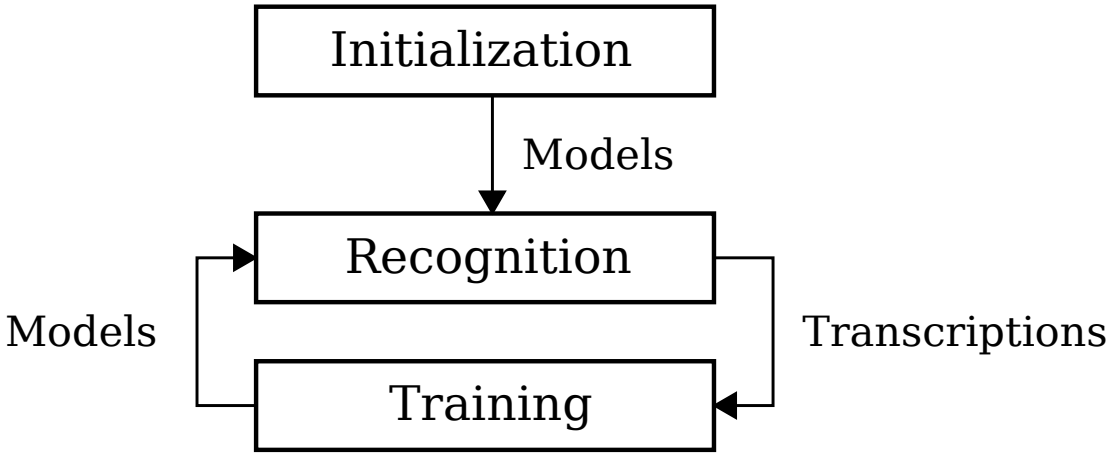


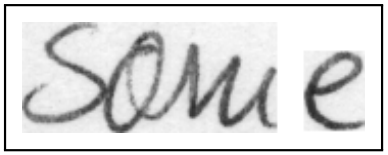
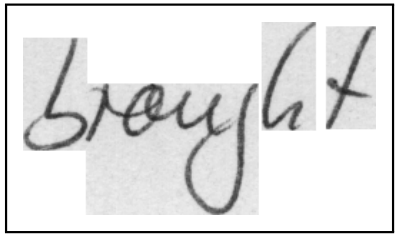
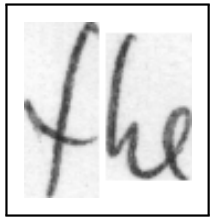
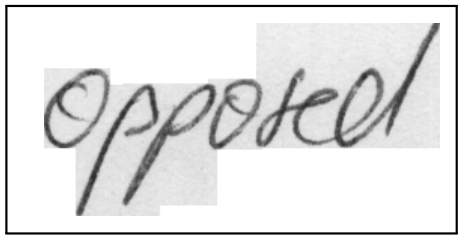
Figure : Illustration of the unsupervised training procedure.

Lexicon construction

- Replace unseen characters by generic **gap model** during decoding
- Guess rare characters by probabilistic constraints encoded in the language model

	sequence of characters					
Initialization	*	*	*	*	*	*
1st iteration	*	e	*	*	r	e
2nd iteration	*	e	*	o	r	e
3rd iteration	b	e	f	o	r	e

Depiction of the training

Image		Initialization	1st iter.	2nd iter.
	=	"this"	"this"	"such"
	=	"before"	"thought"	**"brought"**
	=	**"the"**	**"the"**	**"the"**
	=	"American"	"thought"	"prepared"

Search space

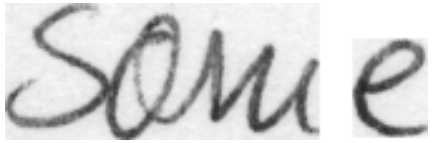
Because the models are weak in the beginning, the state pruning threshold has to be significantly increased or even disabled.

There are several methods to decrease the combinatorical explosion of the search space:

- Unigrams LM only (search space exponential in the order).
- Smaller vocabulary (search space is polynomial in the size).
- Reduce the length of a feature sequence.

Databases

IAM



English handwritten isolated words
Selection of 46k word images for training
7k word images for validation
44k vocabulary size (10k in training)
Unigram word LM, 5% OOV rate

Results

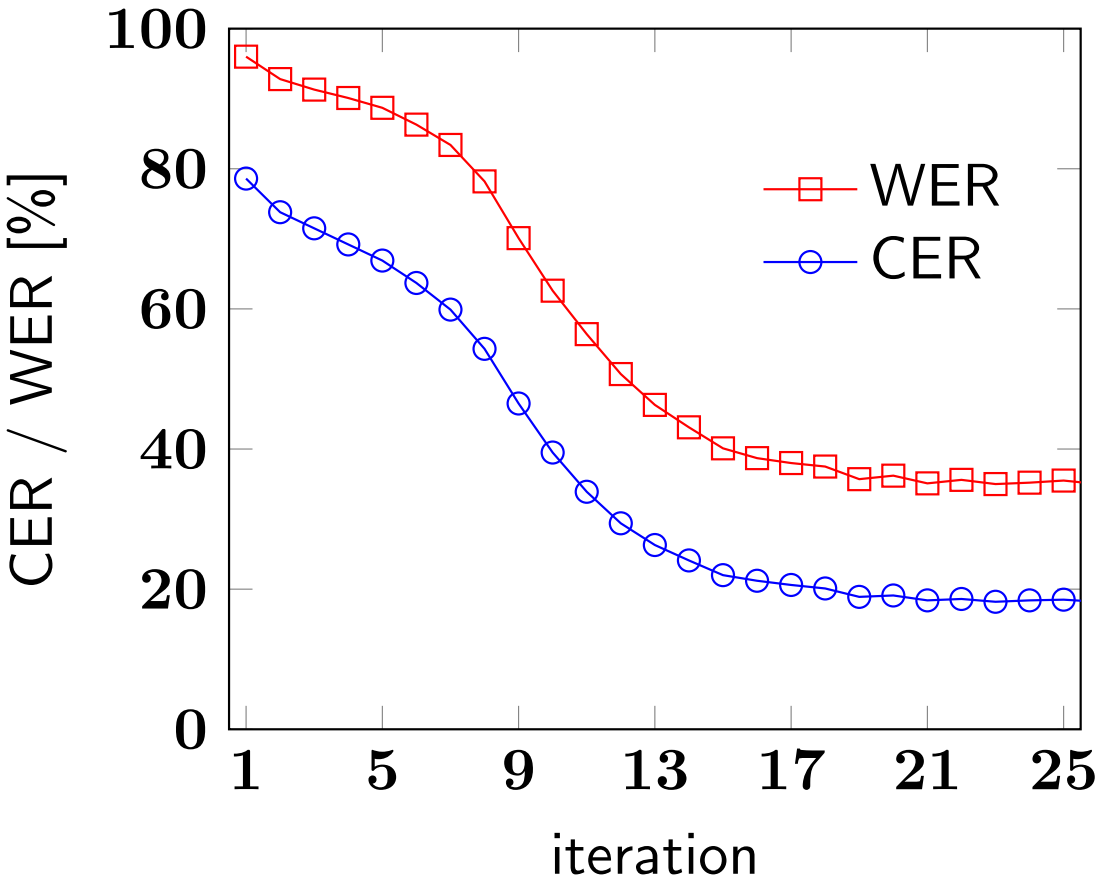
The performance of the unsupervised-trained system is very close to the system trained with labels.

Table : Results on a dataset for English handwritten words (IAM).

	train [%]		dev [%]	
	CER	WER	CER	WER
Supervised	7.5	14.9	9.7	20.5
Unsupervised	15.8	30.7	13.7	28.6

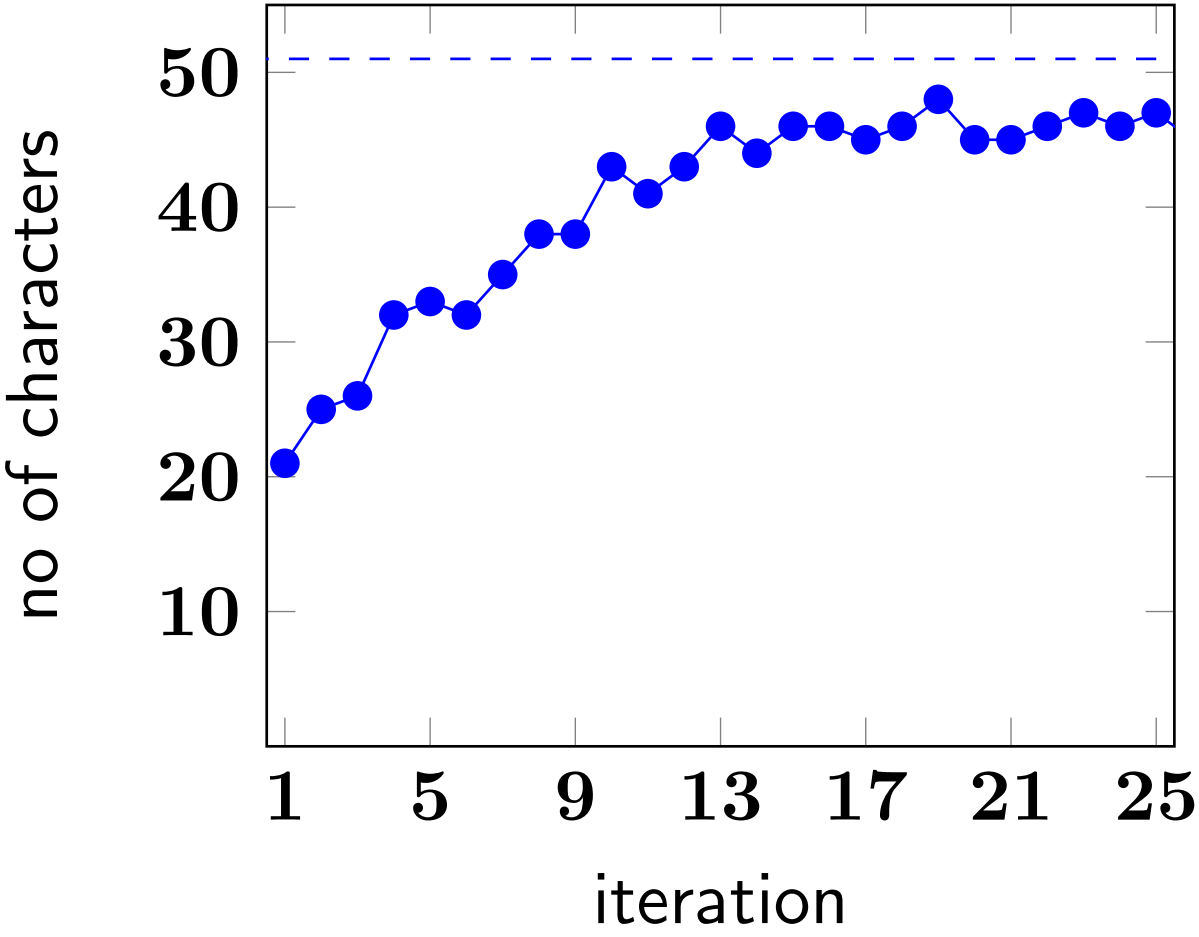
The development set was unseen to both systems. The unsupervised training method has no tendency to overfit.

Results



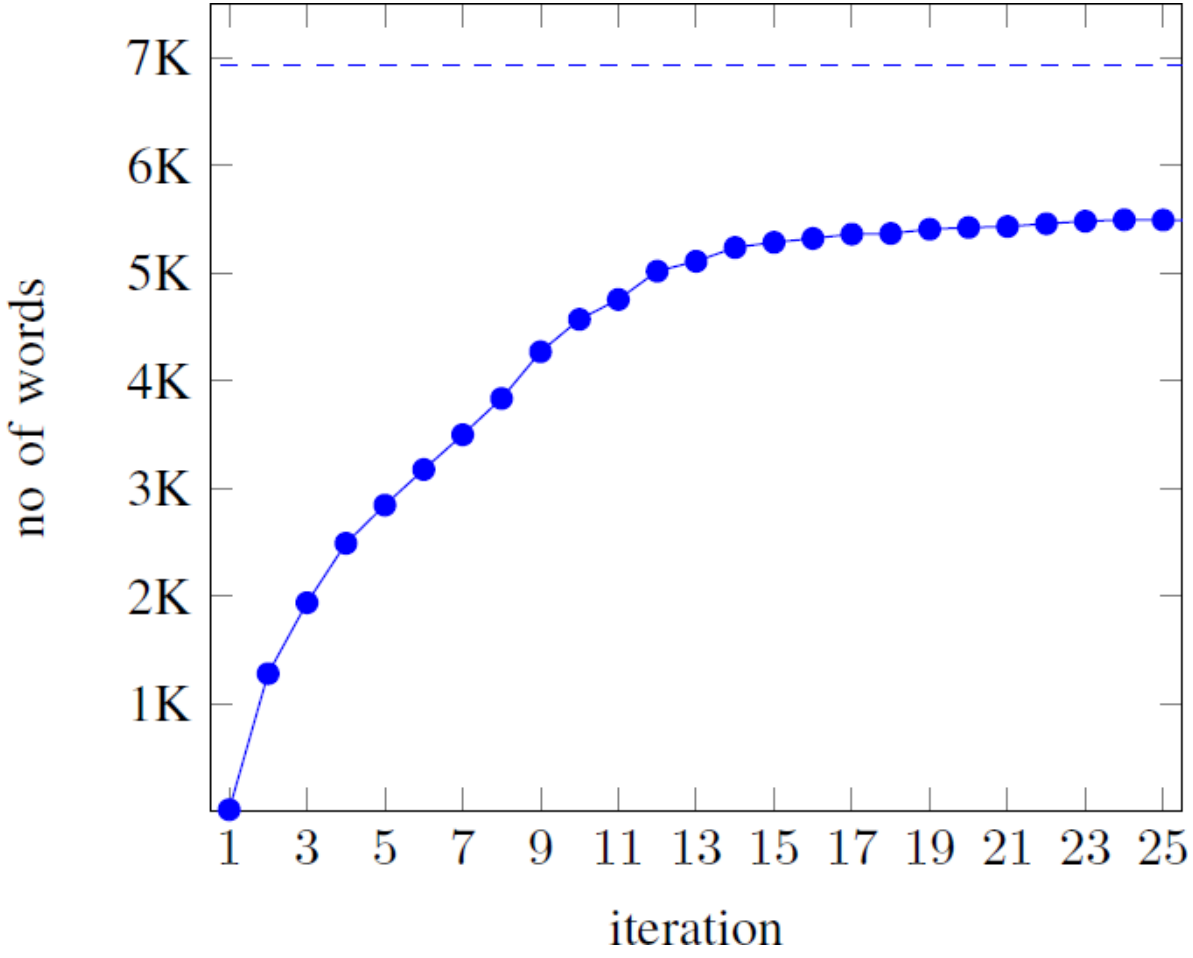
Results

The system was able to learn almost all characters.



Results

The system was able to learn almost all words in the vocabulary.



Conclusions

The performance of a system trained in the unsupervised fashion is close to the one of a system trained with a perfect transcription.

- We use only a prior language model and no annotations of the images.
- The segmentation of words into characters is not provided but uncovered by the system itself.

$\Rightarrow$ The unsupervised approach can be used as aligner

Future work

- Full text line images / higher n-gram language models
- Reduction of approximations
- Investigation of convergence behavior and initialization procedure
- Combination with other classification models (neural networks, etc.)

Thank you for your attention

doetsch@i6.informatik.rwth-aachen.de