Deep Learning Feature for Handwritten Keyword Spotting

Baptiste Wicht Andreas Fischer Jean Hennebert

iCoSys, University of Applied Sciences of Western Switzerland HES-SO

DIVA Group, University of Fribourg, Switzerland

Who's who

Deep Learning Feature for Handwritten Keyword Spotting

Baptiste Wicht
Deep Feature Extraction

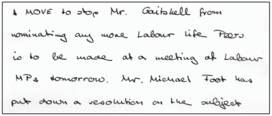
Andreas Fischer
Keyword Spotting

Jean Hennebert
Enjoying Conference

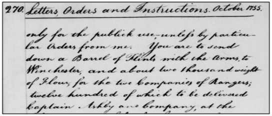
Table of Contents

Introduction - Research Questions

- Are deep learning features good for keyword spotting applications?
- Sub-questions:
  - Are such features robust for different systems?
    - template-based (DTW)
    - learning-based (HMM)
  - Does it work across very different handwritten inputs, i.e. historical 13th century docs to modern English handwriting?
  - Are such features better than state-of-the-art hand-crafted features?
  - How much cooking to get decent performances?


A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject

(a) IAM database

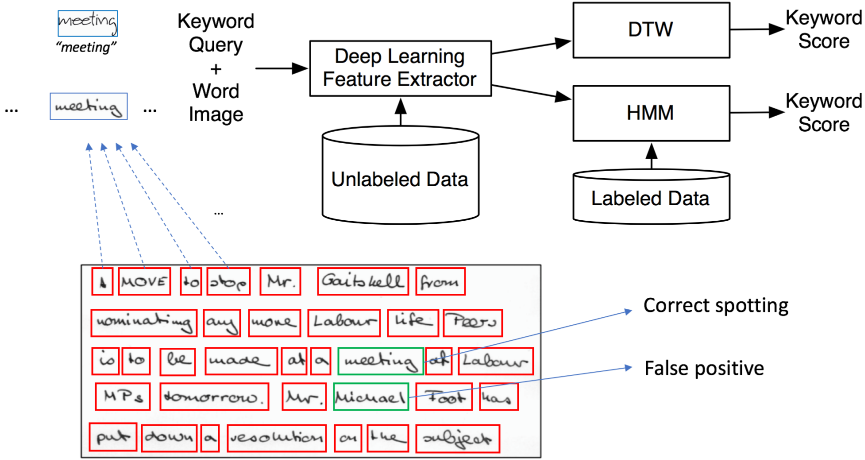

270. Letters Orders and Instructions. October 1755.
only for the publick use, unless by particu-
lar Orders from me. You are to send
down a Barrel of Flints with the Arms, to
Winchester, and about two thousand weight
of Flour, for the two Companies of Rangers;
twelve hundred of which to be delivered
Captain Ashby and Company, at the

(b) George Washington database


machet trurich mir den lip.
dar also manigiu heizet wip.
ir stimme sint geliche hel.
genuge sint gein valsche snel.
etsliche valsches lære.
sus teilent sich diu mære.
daz die geliche sint genamt.

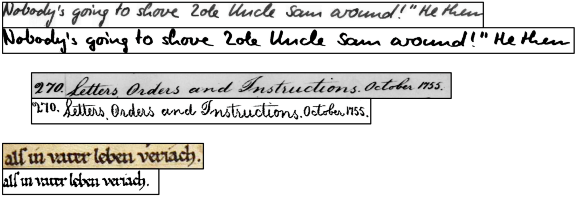
(c) Parzival database

Introduction - Keyword Spotting System

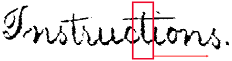


Preprocessing

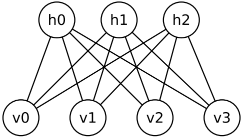
1. The system operates on segmented word images
   - binarized, normalized to remove the skew and slant
   - resized to a third of their height



2. Patches are extracted using an horizontal sliding window
   - no vertical overlap
   - move from left to right one pixel at a time

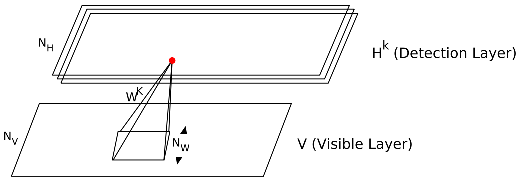


Restricted Boltzmann Machine



- Generative Stochastic Artificial Neural Network (ANN)
- Learn probability distribution over the inputs
- Trained with Contrastive Divergence
  - Similarly to gradient descent techniques
  - As an autoencoder
- Can reconstruct the features (h) from the input (v)
  - And the other way around

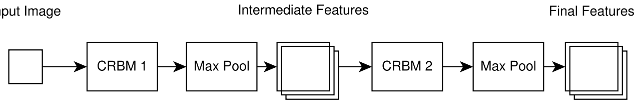
Convolutional RBM



- The layers are connected by convolution
- Input and outputs are matrices
  - 2D Image with C channels as input
  - K 2D feature maps as output
  - $N_W \times N_W$ pixels per patch
  - $[C \times K \times N_W \times N_W]$ weights
- The training principles are the same as for the RBM

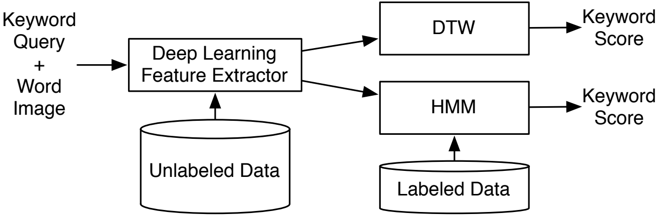
Feature Extractor

- Two CRBM are stacked to form a Convolutional Deep Belief Network



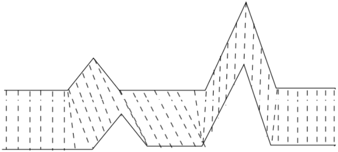
- Max Pooling after each CRBM
  - To improve robustness of features
  - To reduce the number of features
- Normalization of the final features
  - Each feature group is one-sum normalized
  - Each feature is zero-mean and unit variance normalized

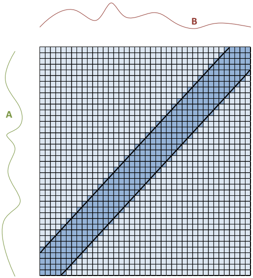
Word Spotting System



- Input:
  - A "target" keyword image K
  - A "candidate" word image X
- Decision: Does the candidate image matches with the keyword ?
  - Decided with a dissimilarity measure and a threshold
  - If $ds(K, X) < T$ then accept the candidate X

Dynamic Time Warping (DTW)



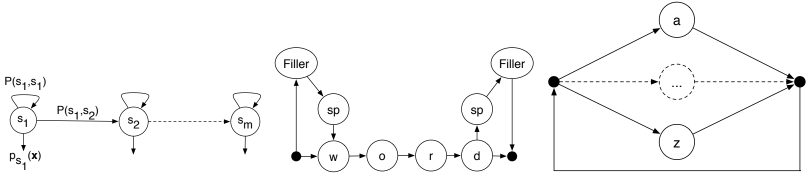


- Find an optimal alignment between two sequences of different length
  - Warped non-linearly to match each other
  - The cost of an alignement is the sum of the distances of aligned pairs
    - Normalized w.r.t. the warping path
- Sakoe-Chiba band is used to improve the results
  - Constrain the search within a band around the shortest path

Source: Wikimedia

Hidden Markov Model (HMM)

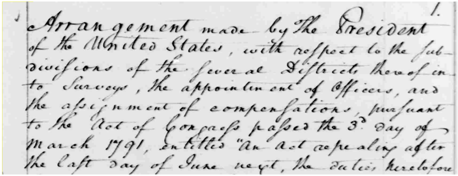
- Based on: Fischer et al. "HMM-based word spotting in handwritten documents using subword models", ICPR 2010

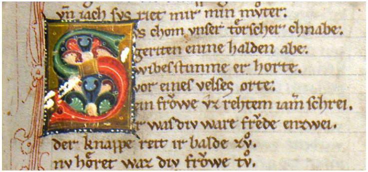


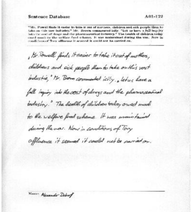
1. One m-state HMM per character, left-right topology
2. Keyword model K is created by connecting character HMMs
3. A filler model F (unconstrained) is created in the same way

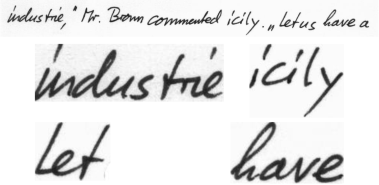
- The dissimilarity is computed with both log-likelihoods measures
  - $ds(X, K) = \frac{\log p(X|F) - \log p(X|K)}{L_k}$

Experimental Evaluation









- Evaluated on three datasets
  - GW: 4894 word images, 1755, English, single-writer
  - PAR: 23485 word images, 13th Century, ancient German, single-writer
  - IAM: 70871 word images, modern English, multiple-writer

Experimental Evaluation

- Evaluated against three baselines
  - `Marti2001`: 9 heuristic features per column of the image
  - `Rodriguez2008`: local gradient histogram features (128-dimensional)
  - `Terasawa2009`: slit-style Histogram Of Gradients (HOG) features (384-dimensional)
- Performance is assessed using two measures:
  - Average Precision (AP): one global threshold
  - Mean Average Precision (MAP): one threshold per keyword
- The number of filters is the only parameter tuned for each data set
  - **All other parameters are kept the same under all configurations**
- Parameters of the classifiers are the same for all systems
  - Taken from: Fischer et al. "HMM-based word spotting in handwritten documents using subword models", ICPR 2010

DTW Results

	GW		PAR		IAM	
System	AP	MAP	AP	MAP	AP	MAP
Marti2001	33.24	45.26	50.67	46.78	5.10	13.57
Rodriguez2008	41.20	63.39	55.82	47.52	00.80	09.73
Terasawa2009	43.76	64.80	69.10	**73.49**	00.56	09.55
Proposed	**56.98**	**68.64**	**72.71**	72.38	1.04	10.27
Relative Improvement	23.20%	5.59%	4.96%	−1.53%	-	-

- Results
  - Better on GW than all the baselines
  - Comparable perf on PAR with best baseline (Terasawa2009)
  - IAM results can be ignored
    - DTW template matching is failing with different writing styles

HMM Results

	GW		PAR		IAM	
System	AP	MAP	AP	MAP	AP	MAP
Marti2001	48.80	69.42	69.47	77.98	16.67	49.24
Rodriguez2008	32.60	59.40	25.43	32.53	5.47	21.11
Terasawa2009	68.01	79.49	90.50	90.53	59.66	71.59
Proposed	**71.21**	**85.06**	**92.34**	**94.57**	**64.68**	**72.36**
Relative Improvement	4.49%	6.54%	1.99%	4.27%	7.76%	1.06%

- Outperforms every baseline in all tested situations

System Optimization

- Optimization of the system has been challenging
  - Large number of parameters
  - Rather different datasets
- Training parameters
  - 25 epochs of Contrastive Divergence
  - **Sparsity for binary units**
- Architecture parameters
  - Two-layer models proved best
  - Sliding window of **20 pixels width**
  - **Number of filters**: 8 (GW) and 12 (PAR/IAM)
    - Very important for DTW
  - Units: Binary (GW) and ReLU (PAR/IAM)

Conclusion

- Proposed system outperforms 3 baselines on 3 data sets
  - Robust performance under all tested conditions
  - With purely unsupervised feature learning
  - Improvements on two different classifiers: DTW and HMMs
- Optimizing the model is non-trivial
  - Large number of parameters
  - DTW is “constraining” about the features
  - Still room for improvement

Future Work - Implementation

Future works

- Use grayscale normalized images
- Augment dataset with distortions
- Find a better configuration specific for HMM
- Score words with potentially better classifiers such as LSTM
- Compare with other auto-encoder types

Implementation

- Freely available online
- Keyword Spotting System (kws), C++
  - `https://github.com/wichtounet/word_spotting`
- Deep Learning Library (DLL), C++
  - `https://github.com/wichtounet/dll`
- *URLs present in the paper*

Questions

Questions ?