CITIlab ARGUS for Keyword Search in Historical Handwritten Documents

Thanks to Mauricio for giving the talk.

Tobias Strauß
Tobias Grüning
Gundram Leifert
Roger Labahn
Handwritten Scanned Document Retrieval Task 2016

- advanced keyword spotting task

<table>
<thead>
<tr>
<th></th>
<th>#pages</th>
<th>#lines</th>
<th>#characters</th>
<th>polygons</th>
<th>baselines</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td>363</td>
<td>9645</td>
<td>65488</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>development set</td>
<td>433</td>
<td>10589</td>
<td>80758</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>test set</td>
<td>200</td>
<td>6355</td>
<td>—</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
System Overview

- **Image**
- **LinInfos**

**Polygon2BaseLine**

**Baseline2Polygon**

**Preprocessing**

**Feature Generation**

**MDRNN**

**Training**

**Decoding**

- **external input**
- **generation of consistent surrounding polygons**
- **optical model (generation of confidence matrices)**
- **advanced KWS (BBs)**
Polygon2BaseLine

BaseLine2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

external input

generation of consistent surrounding polygons

optical model (generation of confidence matrices)

advanced KWS (BBs)
Polygon2BaseLine

1. Convolving the input image with Sobel-Kernels:

Contribute to exempt him from a burden, or a pun
ishment; or to the State only, as an instrument

2. Evaluation of statistics on the sobel image to calculate the baselines

Module Output (Baselines) is calculated given the polygons
Polygon2BaseLine

Baseline2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

External input

Generation of consistent surrounding polygons

Optical model (generation of confidence matrices)

Advanced KWS (BBs)
Baseline2Polygon

1. Work on the sobel image
2. For each baseline calculate a medial seam (by maximizing sobel activity)
3. For each medial seam calculate two separating seams by using dynamic programming to calculate an optimal path from "left to right"
4. Merge both separating seams to get a surrounding polygon per baseline

Module Output (Polygons) is calculated given the baselines
Polygon2BaseLine

BaseLine2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

external input

generation of consistent surrounding polygons

optical model (generation of confidence matrices)

advanced KWS (BBs)
Preprocessing

**Input:** Line image with surrounding polygons

**Image normalization:** contrast enhancement and global size normalization

**Writing normalization:** deskewing and deslanting

**Output:** Normalized line image to fixed image height with mainbody centered in the middle of the image
Polygon2BaseLine

BaseLine2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

external input

generation of consistent surrounding polygons

optical model (generation of confidence matrices)

advanced KWS (BBs)
Feature Generation

**Input:** Normalized line image

**Output:** 3-dimensional frequency feature
Polygon2BaseLine

BaseLine2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

external input

generation of consistent surrounding polygons

optical model (generation of confidence matrices)

advanced KWS (BBs)
Multidimensional Recurrent Neural Network

INPUT LAYER

FEATURE LAYER

2D LAYER 1

Subsample 1

Subsample 2

Subsample 3

2D LAYER 2

0D LAYER 2

OUTPUT LAYER

Contribute to exempt him from a rule, then or a bus.
Output of the Network: Confidence Matrix (ConfMat)

**Input:** 3-dimensional frequency feature

**Output:** Confidence of a character at a specific position (high confidence = black)

Transcription: “contribute to exempt him from a burthen or a biu-”

September 1, 2016
Polygon2BaseLine

BaseLine2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

Image

LinInfos

external input

generation of consistent surrounding polygons

optical model (generation of confidence matrices)

advanced KWS (BBs)
Network Training

- The network is trained using CTC as error function and an extension of Nesterov’s Accelerated Gradient Descent with momentum $\mu = 0.9$ for parameter update.
- One epoch contains 10,000 training samples (lines) from the training set. No lines of the devel or the test set were used for training at all.
- A learning rate $\lambda = 5e^{-4}$ is used for 19 epochs using the original images.
- For 3 epochs noise to preprocess parameter and network activation are added with the same learning rate.
- For additional 19 epochs we set $\lambda = 5e^{-5}$ and add degradation (pixel noise, blur, cross outs,...) to the image.
Polygon2BaseLine

BaseLine2Polygon

Preprocessing

Feature Generation

MDRNN

Training

Decoding

external input

generation of consistent surrounding polygons

optical model (generation of confidence matrices)

advanced KWS (BBs)
Decoding I

Find words regions

For any keyword \( \omega \) and any line image \( X \)

1. Let \( r(\omega) := .* \omega .* \)
2. Find the most likely path \( \pi^* \) matching \( r(\omega) \)
3. Calculate the likelihood

\[
P_{s:e}(\pi^*(\omega) | X) := \prod_{t=s}^{e} y_{t,\pi^*_t}
\]

of \( \omega \). \((s, e)\) are determined by the subpath \( \pi_{s:e} \) which is related to \( \omega \).

4. Accept \( \omega \) if \( P_{s:e}(\pi^*(\omega) | X) \) exceeds a certain threshold
Decoding II

Hyphens

ConfMats of interest end on hyphen symbols such as \(-\):=

1. Find ConfMats matching \( r := \ast [A-Za-z]+ [-:=] \)
2. Combine those columns of the ConfMat which match \([A-Za-z]+\) with the ConfMat of the next line
3. Search in the resulting new ConfMat for keywords
Score

Reestimate the likelihood of the neural net by:

\[
s(\omega) := \frac{1}{N} \frac{P_T(\omega)}{P_S(\omega)} P_{s:e}(\pi^*(\omega) \mid X)
\]

\[
N := \sum_{\omega' \in A^*} \frac{P_T(\omega')}{P_S(\omega')} P_{s:e}(\pi^*(\omega') \mid X)
\]

where \(P_T(\omega)\) is the uni-gram prior probability of \(\omega\) in the corpus and \(P_S(\omega)\) is the prior probability estimated by the network.
Multi word query $\omega_{1:n} := \omega_1, \ldots, \omega_n$

Search all words $\omega_i$ individually and combine their score by

$$s(\omega_{1:n}) := \left( \prod_{i=1}^{n} s(\omega_i) \right)^{\frac{1}{n}}.$$
Decoding V

Implicit network priors $P_S$

Estimate the network priors $P_S$ by

- $\text{abs}$: the real uni-grams $P_S(z) = P_T(z)$
- $\text{prior}$: a constant $P_S(z) \propto 1$
- $\text{da}$: character priors $P_S(z) \propto \left( \prod_{i=1}^{\lvert z \rvert} P(z_i) \right)^c$

where $P(z_i)$ is the character prior and $0 < c \leq 1$.

Results: global average precision (gAP) on segment level

<table>
<thead>
<tr>
<th></th>
<th>abs</th>
<th>prior</th>
<th>da</th>
<th>baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>devel set</td>
<td>94.81</td>
<td>95.36</td>
<td>94.99</td>
<td>74.2</td>
</tr>
<tr>
<td>test set</td>
<td>33.89</td>
<td>43.20</td>
<td>46.74</td>
<td>14.4</td>
</tr>
</tbody>
</table>
Decoding VI

CTC vs. path probability

Compare path probability $P_{s:e}(\pi^*(\omega) \mid X)$ with CTC-probability i.e. the sum over all feasible paths related to $\omega$ in the range from $s$ to $e$.

Results: gAP on segment level

<table>
<thead>
<tr>
<th></th>
<th>abs</th>
<th>prior</th>
<th>da</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>path</td>
<td>ctc</td>
<td>path</td>
</tr>
<tr>
<td>devel set</td>
<td>94.81</td>
<td>94.89</td>
<td>95.36</td>
</tr>
<tr>
<td>test set</td>
<td>33.89</td>
<td>33.98</td>
<td>43.20</td>
</tr>
</tbody>
</table>
Thank you for your attention.

Sorry for not being here.