UNIVERSITAT POLITÈCNICA DE VALÈNCIA

MIARFID Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

DSIC DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Session 1. CRF applications: CRFSuite

Aplicaciones de Reconocimiento de Formas (ARF)

Curso 2023/2024

Departamento de Sistemas Informáticos y Computación

Index

Index

CRF basics

CRF definitions (linear case):

- Input: $x = x_1, x_2, \ldots, x_T \in \mathcal{X}^*$
- Output: $y = y_1, y_2, \ldots, y_T \in \mathcal{Y}^*$
- Model parameters (vector of feature weights of dimension K):

  $\theta = (\theta_1, \theta_2, \ldots, \theta_K) \in \mathbb{R}^K$
- (Bigram) indicator feature: $f_k(y_{t-1}, y_t, x_t)$
- Vector of features (dimension K):

  $f(y_{t-1}, y_t, x_t) = (f_1(y_{t-1}, y_t, x_t), f_2(y_{t-1}, y_t, x_t), \ldots, f_K(y_{t-1}, y_t, x_t))$

CRF basics

- Compatibility function: $\phi(x, y; \theta) \rightarrow \mathbb{R}$

$$\phi(x, y; \theta) = \sum_{t=1}^{T} \theta f(y_{t-1}, y_t, x_t) = \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t)$$

- Conditional distribution:

$$p(y|x; \theta) = \frac{\exp\{\phi(x, y; \theta)\}}{\sum_{y' \in \mathcal{Y}} \exp\{\phi(x, y'; \theta)\}}$$

Final form of the **linear chain CRF** model for $p(y|x)$:

$$p(y|x; \theta) = \frac{\exp\{\phi(x, y; \theta)\}}{Z(x; \theta)} = \frac{\exp\{\sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t)\}}{Z(x; \theta)}$$

Where $Z(x; \theta) = \sum_{y' \in \mathcal{Y}^*} \exp\{\sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y'_{t-1}, y'_t, x_t)\}$

CRF basics

General case for CRF: from linear graph to general graph

- Factor graph: G
- Inputs/outputs according to factors in G: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_A\}$, $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_A\}$
- Model parameters (vectors of feature weights for each factor a, dim. $K(a)$):

$$\theta_a = (\theta_{a1}, \theta_{a2}, \ldots, \theta_{aK(a)})$$

- Indicator feature for a given factor a:

$$f_a(\mathbf{y}_a, \mathbf{x}_a) = (f_{a1}(\mathbf{y}_a, \mathbf{x}_a), f_{a2}(\mathbf{y}_a, \mathbf{x}_a), \ldots, f_{aK(a)}(\mathbf{y}_a, \mathbf{x}_a))$$

CRF basics

- Set of factors in G: $F = \{\Psi_1, \Psi_2, \ldots, \Psi_A\}$

$$\Psi_a(\mathbf{y}_a, \mathbf{x}_a) = \exp\left\{\sum_{k=1}^{K(a)} \theta_{ak} f_{ak}(\mathbf{y}_a, \mathbf{x}_a)\right\}$$

Final form of the **general CRF** model for $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\Psi_a \in F} \exp\left\{\sum_{k=1}^{K(a)} \theta_{ak} f_{ak}(\mathbf{y}_a, \mathbf{x}_a)\right\}$$

CRF basics

Derived problems:

- Inference: $p(y|x;\theta)$, $p(\mathbf{y}|\mathbf{x};\theta)$
- Decoding (tagging): $\text{argmax}_{y\in\mathcal{Y}^*}\, p(y|x;\theta)$, $\text{argmax}_{\mathbf{y}\in\mathcal{Y}}\, p(\mathbf{y}|\mathbf{x};\theta)$
- Parameter estimation (training): $\hat{\theta}$ according to some criterion
  - Maximum likelihood
  - Stochastic gradient
  - Pseudolikelihood
  - Belief propagation
  - Markov Chain Monte Carlo

More information: An Introduction to Conditional Random Fields (C. Sutton & A. McCallum)

Index

CRFSuite

http://www.chokkan.org/software/crfsuite/
https://github.com/chokkan/crfsuite

- Developed by Naoaki Okazaki
- Fast training and tagging
- Simple data format
- Different training methods
- **Linear-chain CRF**
- Performance evaluation on training
- Efficient file format for CRF models

CRFSuite

File format:

- Data file: set of items separated by blank line
- Items: set of lines
- Lines: `label` TAB `attributes`
- Attributes: `attribute=value ... attribute=value`

```
B-NP    w[0]=An w[1]=A.P. pos[0]=DT pos[1]=NNP  __BOS__
I-NP    w[-1]=An  w[0]=A.P. w[1]=Green  pos[-1]=DT  pos[0]=NNP  pos[1]=NNP
I-NP    w[-1]=A.P.  w[0]=Green  w[1]=official pos[-1]=NNP pos[0]=NNP  pos[1]=NN
I-NP    w[-1]=Green w[0]=official w[1]=declined pos[-1]=NNP pos[0]=NN pos[1]=VBD
B-VP    w[-1]=official  w[0]=declined w[1]=to pos[-1]=NN  pos[0]=VBD  pos[1]=TO
I-VP    w[-1]=declined  w[0]=to w[1]=comment  pos[-1]=VBD pos[0]=TO pos[1]=VB
I-VP    w[-1]=to  w[0]=comment  w[1]=on pos[-1]=TO  pos[0]=VB pos[1]=IN
B-PP    w[-1]=comment w[0]=on w[1]=the  pos[-1]=VB  pos[0]=IN pos[1]=DT
B-NP    w[-1]=on  w[0]=the  w[1]=filing pos[-1]=IN  pos[0]=DT pos[1]=NN
I-NP    w[-1]=the w[0]=filing w[1]=.  pos[-1]=DT  pos[0]=NN pos[1]=.
O       w[-1]=filing  w[0]=.  pos[-1]=NN  pos[0]=.  __EOS__

B-NP    w[0]=The  w[1]=$  pos[0]=DT pos[1]=$  __BOS__
I-NP    w[-1]=The w[0]=$  w[1]=40-a-share pos[-1]=DT  pos[0]=$  pos[1]=JJ
I-NP    w[-1]=$ w[0]=40-a-share w[1]=proposal pos[-1]=$ pos[0]=JJ pos[1]=NN
I-NP    w[-1]=40-a-share  w[0]=proposal w[1]=values pos[-1]=JJ  pos[0]=NN pos[1]=VBZ
B-VP    w[-1]=proposal  w[0]=values w[1]=the  pos[-1]=NN  pos[0]=VBZ  pos[1]=DT
B-NP    w[-1]=values  w[0]=the  w[1]=company  pos[-1]=VBZ pos[0]=DT pos[1]=NN
I-NP    w[-1]=the w[0]=company  w[1]=at pos[-1]=DT  pos[0]=NN pos[1]=IN
B-PP    w[-1]=company w[0]=at w[1]=about  pos[-1]=NN  pos[0]=IN pos[1]=RB
B-NP    w[-1]=at  w[0]=about  w[1]=$  pos[-1]=IN  pos[0]=RB pos[1]=$
I-NP    w[-1]=about w[0]=$  w[1]=106.6  pos[-1]=RB  pos[0]=$  pos[1]=CD
I-NP    w[-1]=$ w[0]=106.6  w[1]=million  pos[-1]=$ pos[0]=CD pos[1]=CD
I-NP    w[-1]=106.6 w[0]=million  w[1]=.  pos[-1]=CD  pos[0]=CD pos[1]=.
O       w[-1]=million w[0]=.  pos[-1]=CD  pos[0]=.  __EOS__

B-NP    w[0]=A.P. w[1]=Green  pos[0]=NNP  pos[1]=NNP  __BOS__
I-NP    w[-1]=A.P.  w[0]=Green  w[1]=currently  pos[-1]=NNP pos[0]=NNP  pos[1]=RB
B-ADVP  w[-1]=Green w[0]=currently  w[1]=has  pos[-1]=NNP pos[0]=RB pos[1]=VBZ
B-VP    w[-1]=currently w[0]=has  w[1]=2,664,098  pos[-1]=RB  pos[0]=VBZ  pos[1]=CD
B-NP    w[-1]=has w[0]=2,664,098  w[1]=shares pos[-1]=VBZ pos[0]=CD pos[1]=NNS
I-NP    w[-1]=2,664,098 w[0]=shares w[1]=outstanding  pos[-1]=CD  pos[0]=NNS  pos[1]=JJ
B-ADJP  w[-1]=shares  w[0]=outstanding  w[1]=.  pos[-1]=NNS pos[0]=JJ pos[1]=.
O       w[-1]=outstanding w[0]=.  pos[-1]=JJ  pos[0]=.  __EOS__

B-NP    w[0]=Its  w[1]=stock  pos[0]=PRP$ pos[1]=NN __BOS__
I-NP    w[-1]=Its w[0]=stock  w[1]=closed pos[-1]=PRP$  pos[0]=NN pos[1]=VBD
```

CRFSuite

Attributes:

- They are **arbitrary strings** (although they usually follow the `attribute=value` format)
- Scaling values with `:` separator (`attribute=value:scale`), default 1
- Escape characters: `'\:'`, `'\\'`

File format BNF syntax:

```
<line>          ::= <item> | <eos>
<item>          ::= <label> ('\t' <attribute>)+ <br>
<eos>           ::= <br>
<label>         ::= <string>
<attribute>     ::= <name> | <name> ':' <scaling>
<name>          ::= (<letter> | "\:" | "\\")+
<scaling>       ::= <numeric>
<br>            ::= '\n'
```

CRFSuite

Usual format for attributes:

- Attribute part is:
  - A vector-index format: `w[-2]`, `pos[0]`
  - A vector-index conditioned format: `w[-1]|w[0]`, `pos[-1]|pos[0]|pos[1]`
- The value part is:
  - A single value: `He`, `DT`
  - A conditioned set of values: `reckons|the`, `VBZ|DT|JJ`

Examples:

```
w[-2]=He
pos[0]=DT
w[-1]|w[0]=reckons|the
pos[-1]|pos[0]|pos[1]=VBZ|DT|JJ
```

CRFSuite

Training: `crfsuite learn [OPTIONS] [DATA]`

Options:

- `-t`: type, only `1d` available
- `-a`: algorithm; `lbfgs` (default), `l2sgd`, `ap`, `pa`, `arow`
- `-p`: for model/algorithm parameters (use `-a ALGORITHM -H` to see options)
- `-m`: file to store model (default: not save)
- `-g`: split in `N` groups for cross-validation (`-x`)
- `-x`: use cross-validation
- `-e`: hold out `M` subset in training
- `-h`: help

CRFSuite

Most important parameters (`-p` option):

- For model (`1d`):
  - `feature.minfreq`
  - `feature.possible_states`
  - `feature.possible_transitions`

- For algorithms:
  - `lbfgs`: `c1`, `c2`, `max_iterations`, `epsilon`, `stop`, `delta`
  - `l2sgd`: `c2`, `max_iterations`, `period`, `delta`, `calibration.{eta|rate|samples}`
  - `ap`: `max_iterations`, `epsilon`
  - `pa`: `type`, `c`, `max_iterations`, `epsilon`
  - `arow`: `variance`, `gamma`, `max_iterations`, `epsilon`

CRFSuite

Tagging (decoding): `crfsuite tag [OPTIONS] [DATA]`

Options:

- `-m`: model to be used
- `-t`: give evaluation performance (accuracy, precision, recall, F1)
- `-r`: output reference
- `-p`: output predicted label probability
- `-i`: output predicted label marginal probability
- `-q`: do not output predicted labels
- `-h`: help

Index

Tutorial: text annotation

Based on CRFsuite - Tutorial on Chunking Task

Dataset: CoNLL 2000 shared task

- Dataset for different NLP tasks
- Chunking: assignment of syntactic correlated parts of words
- Features included in the files: word, POS, chunking label

Development:

- Download training and test data from PoliformaT (`01-CRFSuite/Tutorial`)
  Originally available at `https://github.com/teropa/nlp/tree/master/resources/corpora/conll2000`
- Download utility scripts (`crfutils.py`, `chunking.py`) from PoliformaT (same folder)
  Originally available in CRFSuite source code

Tutorial: text annotation

Original data format:

```
...
sharp JJ I-NP
fall NN I-NP
. . O

London JJ B-NP
shares NNS I-NP
closed VBD B-VP
moderately RB B-ADVP
lower JJR I-ADVP
in IN B-PP
thin JJ B-NP
trading NN I-NP
. . O

At IN B-PP
Tokyo NNP B-NP
, , O
...
```

Format expected by CRFSuite:

```
...
I-NP    w[-2]=Monday w[-1]='s w[0]=sharp w[1]=fall ...
I-NP    ... w[0]=fall w[1]=. w[-1]|w[0]=sharp|fall ...
O       ... w[-1]|w[0]=fall|. pos[-2]=JJ ... __EOS__

B-NP    w[0]=London w[1]=shares w[2]=closed ... __BOS__
I-NP    ... w[1]=closed  w[2]=moderately ...
B-VP    ... w[-1]|w[0]=shares|closed ...
B-ADVP  ... pos[-2]=NNS pos[-1]=VBD pos[0]=RB ...
I-ADVP  ... pos[1]=IN pos[2]=JJ pos[-2]|pos[-1]=VBD|RB ...
B-PP    ... pos[-1]|pos[0]=JJR|IN pos[0]|pos[1]=IN|JJ ...
B-NP    ... pos[-2]|pos[-1]|pos[0]=JJR|IN|JJ ...
I-NP    w[-2]=in w[-1]=thin w[0]=trading ...
O       ... pos[-2]=JJ pos[-1]=NN pos[0]=. ... __EOS__

B-PP    w[0]=At w[1]=Tokyo w[2]=, ... __BOS__
B-NP    ... pos[-1]=IN pos[0]=NNP pos[1]=, pos[2]=DT ...
O       ... w[-1]|w[0]=Tokyo|, w[0]|w[1]=,|the ...
...
```

Tutorial: text annotation

Data conversion: `chunking.py` (requires `crfutils.py`)

```
cat train.txt | ./chunking.py > train.crfsuite.txt
cat test.txt | ./chunking.py > test.crfsuite.txt
```

Model training: `crfsuite learn`

```
crfsuite learn -m CoNLL2000.model train.crfsuite.txt
```

Tutorial: text annotation

Expected output:

```
CRFSuite 0.12  Copyright (c) 2007-2011 Naoaki Okazaki

Start time of the training: 2024-02-14T08:41:54Z

Reading the data set(s)
[1] train.crfsuite.txt
0....1....2....3....4....5....6....7....8....9....10
Number of instances: 8937
Seconds required: 4.370

Statistics the data set(s)
Number of data sets (groups): 1
Number of instances: 8936
Number of items: 211727
Number of attributes: 335674
Number of labels: 22

Feature generation
type: CRF1d
feature.minfreq: 0.000000
feature.possible_states: 0
feature.possible_transitions: 0
0....1....2....3....4....5....6....7....8....9....10
Number of features: 452755
Seconds required: 1.620

L-BFGS optimization
c1: 0.000000
c2: 1.000000
num_memories: 6
max_iterations: 2147483647
epsilon: 0.000010
stop: 10
delta: 0.000010
linesearch: MoreThuente
linesearch.max_iterations: 20

***** Iteration #1 *****
Loss: 275528.648286

...

***** Iteration #165 *****
Loss: 13139.375165
Feature norm: 81.074163
Error norm: 2.638386
Active features: 452755
Line search trials: 1
Line search step: 1.000000
Seconds required for this iteration: 0.830

L-BFGS terminated with the stopping criteria
Total seconds required for training: 148.970

Storing the model
Number of active features: 452755 (452755)
Number of active attributes: 335674 (335674)
Number of active labels: 22 (22)
Writing labels
Writing attributes
Writing feature references for transitions
Writing feature references for attributes
Seconds required: 0.980
```

Tutorial: text annotation

Model testing: `crfsuite tag`

Reference and assigned label

```
crfsuite tag -r -m CoNLL2000.model test.crfsuite.txt
```

Expected output:

```
B-NP    B-NP  |  I-NP    I-NP  |  B-NP    B-NP  |  B-VP    B-VP  |  I-NP    I-NP  |  B-PP    B-PP
I-NP    I-NP  |  I-NP    I-NP  |  I-NP    I-NP  |  B-NP    B-NP  |  I-NP    I-NP  |  B-NP    B-NP
I-NP    I-NP  |  B-VP    B-VP  |  B-PP    B-PP  |  I-NP    I-NP  |  B-PP    B-PP  |  I-NP    I-NP
B-NP    B-NP  |  B-NP    B-NP  |  B-NP    B-NP  |  B-VP    B-VP  |  B-NP    B-NP  |  O       O
I-NP    I-NP  |  I-NP    I-NP  |  B-NP    B-NP  |  B-SBAR  B-SBAR|  I-NP    I-NP  |  B-NP    B-NP
I-NP    I-NP  |  B-PP    B-PP  |  I-NP    I-NP  |  B-NP    B-NP  |  O       O     |  I-NP    I-NP
B-VP    B-VP  |  B-NP    B-NP  |  I-NP    I-NP  |  B-VP    B-VP  |                |  B-NP    B-NP
B-NP    B-NP  |  I-NP    I-NP  |  O       O     |  I-VP    I-VP  |  B-NP    B-NP  |  I-NP    I-NP
B-VP    B-VP  |  B-VP    B-VP  |                |  B-NP    B-NP  |  B-VP    B-VP  |  I-NP    I-NP
B-NP    B-NP  |  I-VP    I-VP  |  B-NP    B-NP  |  I-NP    I-NP  |  O       O     |  ...
```

Tutorial: text annotation

Model testing: `crfsuite tag`

Evaluation measures

```
crfsuite tag -qt -m CoNLL2000.model test.crfsuite.txt
```

Expected output:

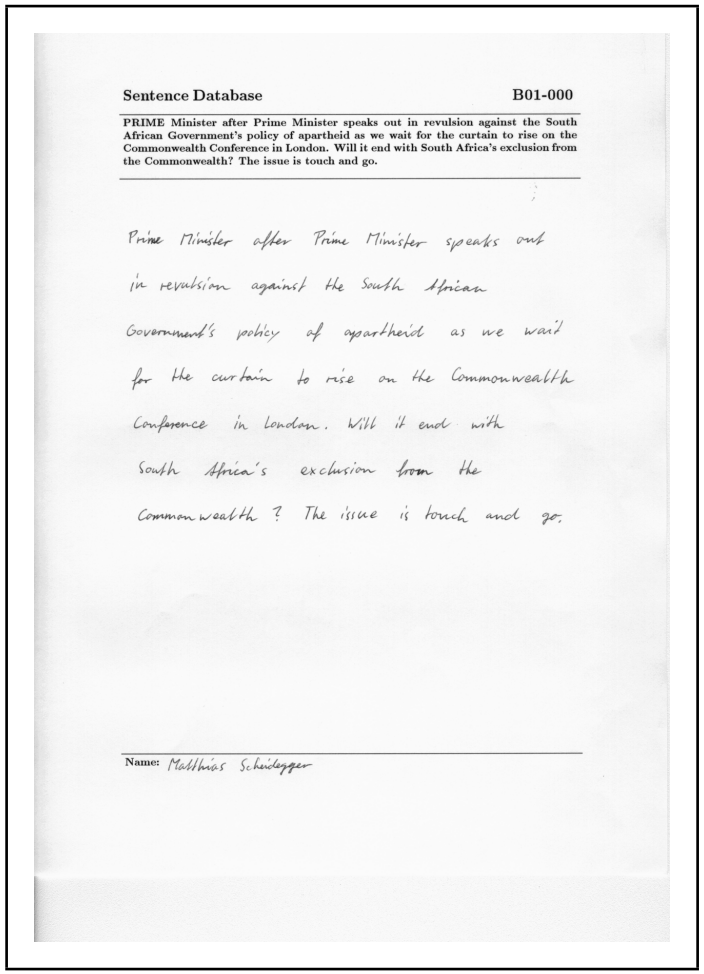
```
Performance by label (#match, #model, #ref) (precision, recall, F1):
    B-NP: (12000, 12358, 12407) (0.9710, 0.9672, 0.9691)
    B-PP: (4707, 4872, 4805) (0.9661, 0.9796, 0.9728)
    I-NP: (13984, 14484, 14359) (0.9655, 0.9739, 0.9697)
    B-VP: (4466, 4662, 4653) (0.9580, 0.9598, 0.9589)
    I-VP: (2549, 2698, 2643) (0.9448, 0.9644, 0.9545)
    B-SBAR: (448, 498, 534) (0.8996, 0.8390, 0.8682)
    O: (5939, 6113, 6174) (0.9715, 0.9619, 0.9667)
    B-ADJP: (322, 403, 438) (0.7990, 0.7352, 0.7658)
    B-ADVP: (711, 835, 866) (0.8515, 0.8210, 0.8360)
    I-ADVP: (54, 82, 89) (0.6585, 0.6067, 0.6316)
    I-ADJP: (110, 137, 167) (0.8029, 0.6587, 0.7237)
    I-SBAR: (2, 15, 4) (0.1333, 0.5000, 0.2105)
    ...
    I-UCP: (0, 0, 0) (******, ******, ******)
Macro-average precision, recall, F1: (0.639239, 0.602512, 0.611086)
Item accuracy: 45422 / 47321 (0.9599)
Instance accuracy: 1176 / 2011 (0.5848)
Elapsed time: 0.670000 [sec] (3003.0 [instance/sec])
```

Index

Practical task: image processing

Proposed task: detection of text lines in handwritten text

Examples:

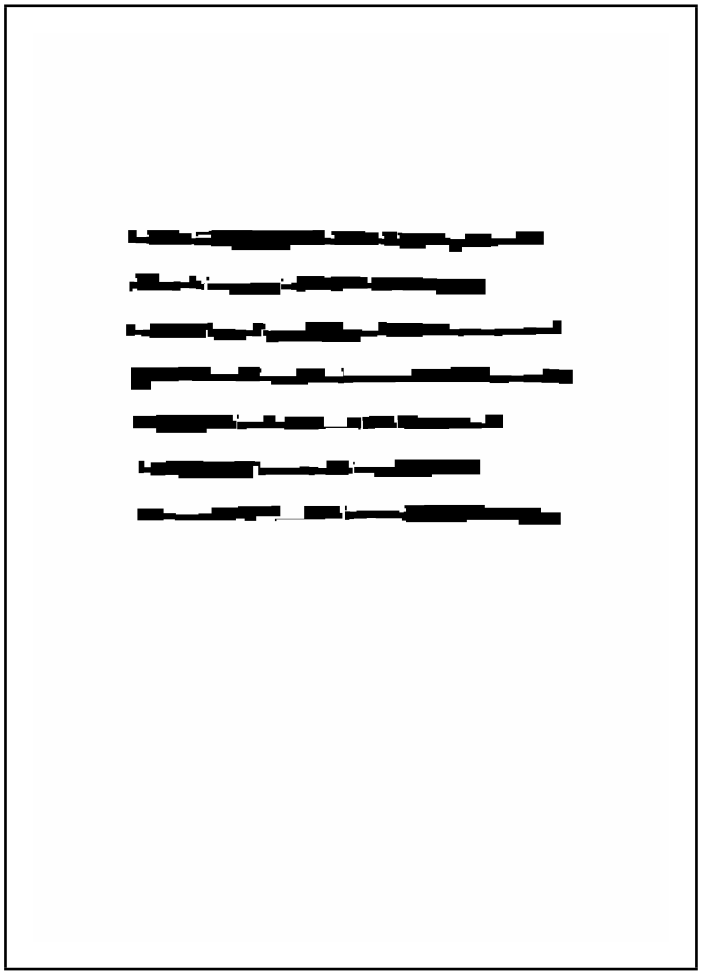


Sentence Database B01-000

PRIME Minister after Prime Minister speaks out in revulsion against the South African Government's policy of apartheid as we wait for the curtain to rise on the Commonwealth Conference in London. Will it end with South Africa's exclusion from the Commonwealth? The issue is touch and go.

Prime Minister after Prime Minister speaks out in revulsion against the South African Government's policy of apartheid as we wait for the curtain to rise on the Commonwealth Conference in London. Will it end with South Africa's exclusion from the Commonwealth ? The issue is touch and go.

Name: Matthias Scheidegger



Practical task: image processing

Dataset publicly available (under registration): IAMDB

Includes images and XML annotations that allow to generate line bodies

Reduced dataset available in PoliformaT (`01-CRFSuite/Practical Task`):

- Pages `b*` for training (114)
- Pages `d*` for testing (82)

Available files: (`.tgz` packages)

- PNG files of the images ($\approx$ 166Mb)
- PNG files with lines bodies ($\approx$ 11Mb)
- All files 25% less size than original images

Practical task: image processing

Proposed task: create features files for CRFSuite and test their performance with the proposed training/test partitions (detecting windows with most pixels part of the line body)

Example of feature extraction:

- Average gray level of 25 $\times$ 25 window
- Average gray level and position (row and column) of 25 $\times$ 25 window

Feature extraction scripts in PoliformaT (`extract_feat.py` and `extract_feat_rc.py`):

- Average gray + row and column
- 25×25 pixel windows
- Features for the current window (no context)

Practical task: image processing

Proposed task:

- Perform proposed feature extraction with training and test sets
- Create CRF model with CRFSuite (`crfsuite learn`) on training data
  - Gray level: About 170 iterations, about 15"
  - Gray level, row, column: About 165 iterations, about 15"
- Test model with CRFSuite (`crfsuite tag`) on test data

  Expected results for gray level:

```
Performance by label (#match, #model, #ref) (precision, recall, F1):
    1: (261950, 282054, 264091) (0.9287, 0.9919, 0.9593)
    0: (6905, 9046, 27009) (0.7633, 0.2557, 0.3830)
Macro-average precision, recall, F1: (0.846022, 0.623774, 0.671147)
```

  Expected results for gray level, row, column:

```
Performance by label (#match, #model, #ref) (precision, recall, F1):
    1: (257622, 264508, 264091) (0.9740, 0.9755, 0.9747)
    0: (20123, 26592, 27009) (0.7567, 0.7450, 0.7508)
Macro-average precision, recall, F1: (0.865349, 0.860276, 0.862790)
```

Practical task: image processing

Proposed task:

- Explore different training option (algorithms, parameters, etc.)
- Explore different feature extraction possibilities:
  - Window size
  - Other features
    - Specific pixel values
    - Gradients of mass
  - Include context
    - Above and below windows
    - Left and right windows
    - Four corner windows
  - Include dependencies