JCDL 2008, Pittsburgh, Pennsylvania

CRF-Based Authors' Name Tagging for Scanned Documents

Manabu Ohta* and Atsuhiro Takasu**

*Okayama University

**National Institute of Informatics

June 19, 2008

Agenda

- Background
  - Motivation of our research
- Two-tier authors’ names extraction
  - Authors’ block extraction
  - CRF-based author/delimiter tagging
- Experiments
- Conclusions and future works

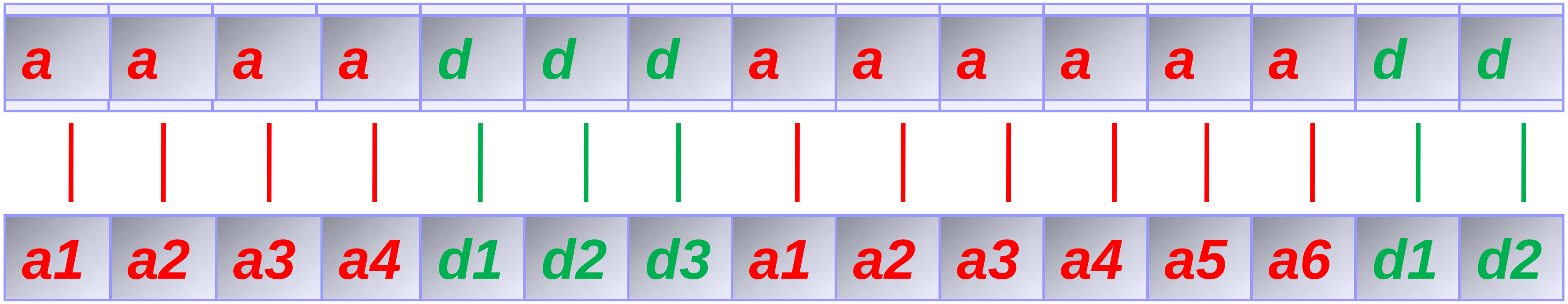
Motivation

- Digitizing process of printed docs for DL
  - Scan => Analyze => Recognize => Store
  - Construction of bibl. DB is labor-intensive
- Automatic extraction of bibl. data from scanned academic articles
  - Cost-effective
  - Need to be error-tolerant to OCR errors
- Why extract “authors’ names”?
  - Because among the most critical bibl. elements

Authors’ names extraction

- Two-tier authors’ names extraction
  - Authors’ block extraction
    - Extract a block representing authors from a title page
  - CRF-based author/delimiter tagging
    - Label every character as either author or delimiter
- Authors’ block example
- Our OCR system
  - English and Japanese OCR engines
  - Layout analysis + character recognition
  - Output bounding rectangles for chars, lines, blocks

Author/delimiter tagging



a	a	a	a	d	d	d	a	a	a	a	a	a	d	d
a1	a2	a3	a4	d1	d2	d3	a1	a2	a3	a4	a5	a6	d1	d2

- Tag sets
  - 2-tag set: mere *author* or *delimiter*
  - 2+pos-tag set: 2-tag set with character positions
    - The max positions of *a* and *d* determined by training

Conditional Random Fields (CRF)

$$\mathbf{y} = t_1 t_2 \cdots t_n \text{ : a tag sequence}$$

$$\mathbf{x} = c_1 c_2 \cdots c_n \text{ : an input char sequence}$$

- □ Our formulation

Normalization factor

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_{i=1}^{n} \sum_{k} \lambda_k f_k(t_{i-1}, t_i, \mathbf{x})\right)$$

Feature function

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in Y(\mathbf{x})} P(\mathbf{y} \mid \mathbf{x})$$

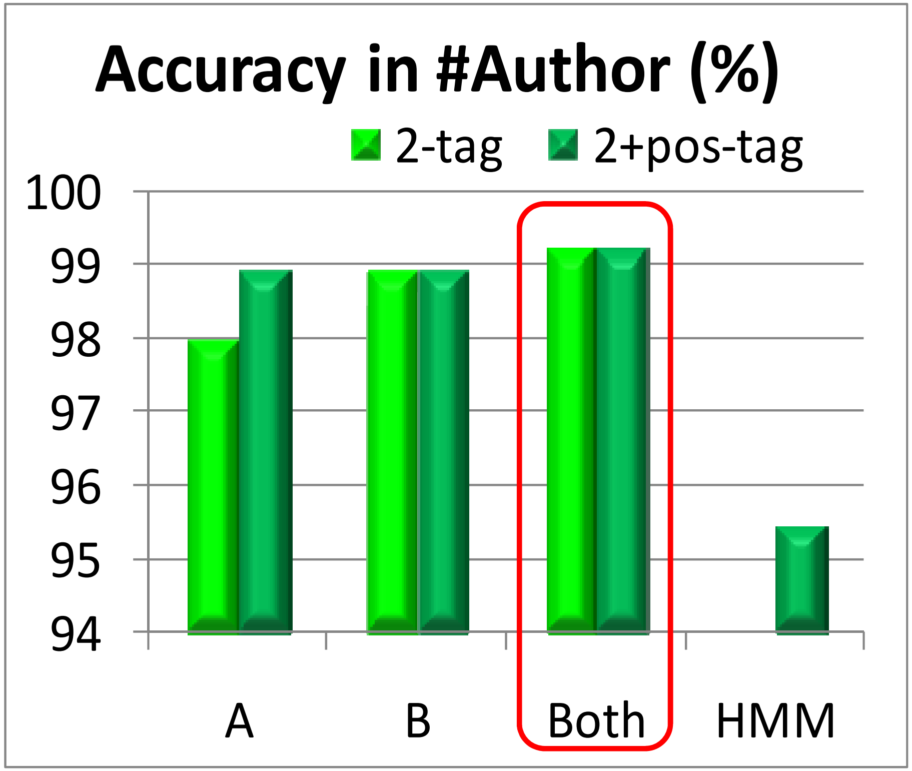
- □ Feature functions
  - Features: chars (textual info.) & widths (layout info.)
  - E.g.

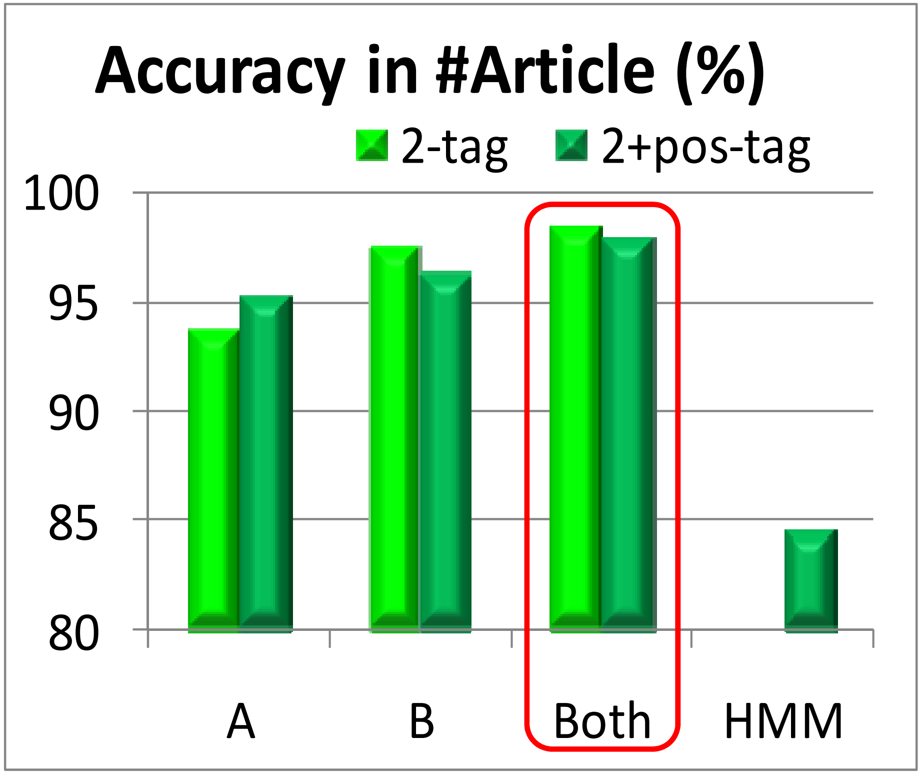
$$f_k(t_{i-1}, t_i, \mathbf{x}) = \begin{cases} 1 & \text{if } c_i = \text{'f'}, t_i = \text{d} \\ 0 & \text{otherwise} \end{cases}$$

Experiments

- Data : OCR-processed academic articles
  - 54 issues of TIPSJ in 2003(vol.44), 2004(vol.45)
  - Training: vol.44, Test: half of vol.45
- OCR accuracy
  - 99.00% for abstract, 97.01% for references
- Implementation: CRF++ 0.50
- Selected features for our CRF
  - <c(0)>: character unigram
  - <w(0)>: character’s width unigram
  - <t(-1),t(0)>: tag bigram

Tagging accuracy (test data)





A: <c(0)>+<t(-1),t(0)> B: <w(0)>+<t(-1),t(0)>

Both: <c(0)>+<w(0)>+<t(-1),t(0)>

HMM: our HMM-based tagger

Discussion

- The setting “Both”
  - Achieved 99.22% accuracy (in #Author)
  - Outperformed A, B, and our HMM-based tagger
- 2-tag vs. 2+pos-tag sets
  - Almost no difference in this experiment
- Tagging errors
  - Caused by OCR errors & noises of documents
  - Often occur at the boundary between name and delimiter strings

Conclusions

- Proposed a CRF-based authors' name tagger
  - Applied it after extracting authors' (text) blocks
  - More than 99% tagging accuracy
  - It outperformed our HMM-based one
- Future works
  - Accuracy improvement with other features
  - Title page analysis system for automatic extraction
  - Extracting other bibl. data such as title, abstract, ...

Questions and Comments?