2012 10th IAPR International Workshop on Document Analysis Systems

# CRF-based Bibliography Extraction from Reference Strings Focusing on Various Token Granularities

Manabu Ohta, Daiki Arauchi
*Okayama University*
*Okayama, Japan*
*Email: {ohta, arauchi}@de.cs.okayama-u.ac.jp*

Atsuhiro Takasu, Jun Adachi
*National Institute of Informatics*
*Tokyo, Japan*
*Email: {takasu, adachi}@nii.ac.jp*

***Abstract*—The references of academic articles include important bibliographic elements such as authors' names and article titles. Automatic extraction of these elements is useful because they can be used for various purposes, including searching. In this paper, a method for automatically extracting bibliographic elements from the text of reference strings is proposed. The proposed method assigns bibliographic labels to reference strings by using linguistic information and conditional random fields. Experimental results indicated that the extraction accuracies of major bibliographies were more than 96%.**

***Keywords*-conditional random field (CRF); bibliography extraction; reference; tokenization; delimiter**

## I. Introduction

In cyber space, the readability of digitized documents is improved by linking them to related ones to generate networked documents. For example, by linking the technical terms appearing in a document to the corresponding dictionary pages on the Internet, readers can check the meaning of the terms efficiently and effectively. Documents such as research papers often contain references, and it is convenient if we can access cited papers without manually searching for them. Some researchers and publishers are trying to build systems that provide direct access to the cited articles. In order to build such systems, automatic bibliography extraction from reference strings is key in terms of reducing the cost of preparing the data. Once bibliographies are extracted, reference entities can be identified by matching against existing bibliographic databases, and these entities can then be linked to the identified papers.

In this paper, we describe our on-going effort to develop a CRF-based bibliography extractor from reference strings. Our particular focus has been on the effect of the various token granularities on the extractor's performance.

## II. Related Work

### A. CRF

A CRF is a statistical framework for modeling sequences that was proposed by Lafferty et al. [1] for part-of-speech (POS) tagging and syntactical analysis. A CRF can outperform other popular models, such as HMMs and maximum entropy models, when the true data distribution has higher order dependencies than those of these other models, which is often the case under practical circumstances [2]. CRFs have performed well in many studies in fields ranging from bioinformatics to natural language processing [3].

In this study, therefore, we apply a common linear-chain CRF to labeling tokens constituting a reference string. That is, we define the conditional probability of a label sequence, $\boldsymbol{y} = y_1, ..., y_n$, given an input-token sequence, $\boldsymbol{x} = x_1, ..., x_n$, as

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp \left\{ \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k f_k(y_{i-1}, y_i, \boldsymbol{x}) \right\}, \quad (1)$$

where $Z(\boldsymbol{x})$ is the normalization constant that makes the probability of all candidate-label sequences sum to one, $f_k(y_{i-1}, y_i, \boldsymbol{x})$ is an arbitrary feature function over $i$th label $y_i$, its previous label $y_{i-1}$, and input sequence $\boldsymbol{x}$, and $\lambda_k$ is a learned weight associated with the feature function, $f_k$.

A CRF assigns the label sequence, $\boldsymbol{y}^*$, to the given-token sequence, $\boldsymbol{x}$, that maximizes Eq. (1), i.e.,

$$\boldsymbol{y}^* := \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{x}). \quad (2)$$

Note that the input-token sequence, $\boldsymbol{x}$, is the sequence of tokens acquired by segmenting a reference string, while the label sequence, $\boldsymbol{y}$, is the sequence of names of bibliographic elements such as titles and authors' names.

### B. Bibliography Extraction from Research Papers

Abekawa et al. proposed an SVM-based method of extracting bibliographies from various research papers in PDF format [4]. They were able to extract 12 types of bibliographic elements from title pages with a 69.2% accuracy and also 6 kinds of bibliographies from reference strings with accuracies of 74.8% (in Japanese) and 81.6% (in English). Peng et al. have also proposed a CRF-based method of extracting bibliographies from the title pages and reference sections of research papers in PDF format [5]. They correctly labeled entire title pages of research papers with a 73.3% accuracy using 13 bibliographic labels defined for title pages, and they correctly labeled 77.3% of reference strings using 13 bibliographic elements defined for references.

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.28

IEEE computer society

Okada et al. [6] proposed a method to automatically extract bibliographic elements, such as authors' names and a title, in a reference string by using SVMs and an HMM. They used both SVMs and the HMM to create an accurate identification of a bibliography for tokens. They reported that they could correctly extract all the bibliographies of 97.6 % of reference strings used for their experiments. Moreover, they empirically showed that their method hardly had a different performance for Japanese and English languages.

We have also previously developed an automatic method of extracting bibliographies from a *title page* of academic articles scanned with *OCR* markup. The method uses a CRF to serially label OCRed text lines on an article's title page as appropriate bibliographic entity names [7], [8].

## III. CRF-based Bibliography Extraction Method

### A. Task Description

In this paper, we propose a CRF-based bibliography extraction method from reference strings of research papers. For example, suppose the following reference string is given:

- M. Ohta, R. Inoue, and A. Takasu, "Empirical evaluation of active sampling for CRF-based analysis of pages," in *Proc. of IEEE IRI 2010*, 2010, pp. 13–18.

Our goal is to extract all the major bibliographic elements (such as authors' names and the title) from it. Hence, we want to generate a bibliographically labeled data set similar to the following:

*<Author>M. Ohta</Author>,*
*<Author>R. Inoue</Author>, and*
*<Author>A. Takasu</Author>,*
*"<Title>Empirical evaluation of active sampling for CRF-based analysis of pages</Title>,"*
*in* Proc. of *<Conf>*IEEE IRI 2010*</Conf>,*
*<Year>2010</Year>,*
*pp. <Pages>13–18</Pages>.*

Our CRF-based bibliography extraction is two-tiered: first it segments each reference string into tokens, and then it labels the tokens in a token sequence with appropriate bibliographic names.

### B. Tokenization

In the proposed method, we first tokenize reference strings by using specifically defined delimiters. In the simplest sense, each character string segmented by any delimiter is a token. Ideally, there should be just one token generated for each bibliographic element.

*1) Tokenization Using Delimiters:* We define as delimiters a comma (,), period (.), space character (_), and double quotation (" ") in addition to the following character strings: *and, eds., ed., (Eds.), editors, No., no., nos., pp., p., Vol., and vol.*. We also define the Japanese counterparts of the above symbols and strings as delimiters. We denote a set of such delimiters as delimiter set 1 and denote a delimiter set 1 without a space character or period as delimiter set 2.

*Tokenization Using Delimiter Set 1:* Tokenization using delimiter set 1 often involves over-segmentation: one bibliographic element is frequently divided into two or more tokens. For example, *B. Obama* is segmented into four tokens—*<t>B</t>*, *<t>.</t>*, *<t>_</t>*, and *<t>Obama</t>*—if delimiter set 1, which includes a period and a space character, is applied.

We used 4,814 reference strings for the experiments described in section IV. Only 15.56 % of them were tokenized without over-segmentation when delimiter set 1 was used. All the English references were over-segmented, whereas in contrast some of the Japanese references were segmented without over-segmentation because Japanese words are not separated by space characters.

*Tokenization Using Delimiter Set 2:* We also prepared a delimiter set 2 by removing the period and space character from delimiter set 1 because handling these characters as delimiters leads to over-segmentation, especially in English references. However, note that these two characters sometimes do work as delimiters for bibliographies. In such a case, delimiter set 2 leads to *under*-segmentation, where two or more bibliographic elements are incorrectly combined into one token. Under-segmentation cannot be compensated for by bibliography labeling afterward because we assign one bibliographic label to each token.

Space characters as delimiters appear in a limited selection of our experimental data set, such as the one between the month and year in "*Jan._2000*". Therefore, we heuristically determined a rule to selectively handle space characters as a delimiter.

We could tokenize 98.46 % of the experimental data in section IV without either over- or under-segmentation by applying delimiter set 2 and a few heuristic rules like the one on a space character.

*2) Tokenization Using B-I-O Tags:* We also propose tokenization by using B-I-O tags [9] to compensate for the over-segmentation caused by *delimiter set 1*. When using delimiter set 1, we first segment reference strings into *words* that are over-segmented tokens. We then prepare B-I-O tags ($\sum = \{RB, RI, DB, DI\}$) and let a CRF assign an appropriate tag to each *word*. Note here that $RB$ and $DB$ denote the first words of the bibliography and the delimiter, respectively, and $RI$ and $DI$ denote the non-initial ones. After labeling, $RB$ and its succeeding $RIs$ (if any) and $DB$ and its succeeding $DIs$ (if any) are combined into respective tokens. Using B-I-O tags thus results in a lower number of tokens than using delimiter set 1 alone. Moreover, B-I-O tags can be used with other journals in which the heuristics of delimiter set 2 do not work well.

When we applied five-fold cross validation to the experimental data in section IV, we could tokenize 91.21 % of them without either over- or under-segmentation.

Table I
BIBLIOGRAPHIC ELEMENT LABELS

| Bibliographic element | Label |
|---|---|
| Author | RA |
| Editor | RE |
| Title | RT |
| Book title | RB |
| Journal | RW |
| Conference | RC |
| Volume | RV |
| Number | RN |
| Page | RPP |
| Publisher | RP |
| Day | RD |
| Month | RM |
| Year | RY |
| Other | O |

Table II
DELIMITER LABELS

| Delimiter | Label |
|---|---|
| . (period) | D |
| “ (double quotation) | DS |
| ,”_ (comma + double quotation + space character) | DE |
| _and_, and_ | DAND |
| eds., eds._, ed., ed._, (Eds.), editors | DED |
| No., no., nos. | DN |
| pp., p. | DPP |
| Vol., vol. | DV |
| , (comma) | DCO |
| ,_ (comma + space character) | DC |
| _ (space character) | DSP |

### C. CRF-based Token Labeling

*1) Token Labeling:* As discussed in section III-A, our CRF-based method labels each token of the inputted reference string with a bibliographic name, such as authors' names. For this purpose, we define a set of bibliographic elements for extraction (Table I) and a set of delimiters to be labeled (Table II). With the delimiters, we also define the Japanese counterparts of DS, DE, and DCO as DZS, DZE, and DZCO, respectively; DED also includes the Japanese counterparts of delimiting strings. The delimiters in Table II are not significant on their own, but they can function as clues for bibliographies to be determined and help to improve overall token labeling accuracy. By using both labels, the CRF-based method labels the sample reference string shown in section III-A as follows.

<RA>M. Ohta</RA>
<DC>, </DC>
<RA>R. Inoue</RA>
<DC>, </DC>
<DAND>and </DAND>
<RA>A. Takasu</RA>
<DC>, </DC>
<DS>“</DS>
<RT>Empirical evaluation of active sampling for CRF-based analysis of pages</RT>
<DE>,”</DE>
<RC>in Proc. of IEEE IRI 2010</RC>
<DC>, </DC>
<RY>2010</RY>
<DC>, </DC>
<DPP>pp. </DPP>
<RPP>13–18</RPP>
<D>.</D>

*2) Features Used for Labeling:* Table III summarizes the set of adopted feature templates that automatically generate a set of feature functions for the token labeling. The features used for the token labeling include the position of a token in a token sequence, the number of the characters grouped by character type that constitute a token, the token character string, and the presence of predefined keywords in preceding, current, and succeeding tokens.

For example, the following is a feature function generated by the unigram feature template <token(0)> when we use delimiter set 2:

$$f_k(y_{i-1}, y_i, \boldsymbol{x}) = \begin{cases} 1 & \text{if } x_i = \text{“M. Ohta”}, y_i = \text{RA} \\ 0 & \text{otherwise} \end{cases} . \tag{3}$$

An example of the feature functions generated by the bigram template <$y$(-1), $y$(0)> is

$$f_k(y_{i-1}, y_i, \boldsymbol{x}) = \begin{cases} 1 & \text{if } y_{i-1} = \text{DS}, y_i = \text{RT} \\ 0 & \text{otherwise} \end{cases} . \tag{4}$$

Such a label bigram reflects the syntactic constraints of reference strings, i.e., that a paper title, RT, is preceded by a double quotation, DS, and precedes a comma and the other double quotation, DE, etc. bearing in mind, of course, that different journals have slightly different layouts.

The number of feature functions generated by the unigram feature template, e.g., <position(0)> is 28 × $N$, where 28 is the number of output classes consisting of 14 kinds of bibliographic elements (Table I) and 14 kinds of delimiters, including Japanese-specific ones (Table II), and $N$ is the number of different token positions in a token sequence. The number of those generated by the bigram feature template <$y$(-1), $y$(0)> amounts to 28 × 28.

## IV. EMPIRICAL EVALUATION

### A. Experimental Setup

The task in the experiment was to extract bibliographic components from the reference strings of research papers. We tested the CRF-based token labeling technique with varying tokenization methods on Japanese academic papers issued by the Institute of Electronics, Information and Communication Engineers in Japan. We used papers issued in 2000 in this experiment. Our dataset consisted of 312 papers and had 4,814 reference strings in total.

Our CRF-based labeling method uses the CRF++ package [10], which is an open source implementation of CRFs for labeling sequential data. When training the CRFs, we set

Table III
FEATURE TEMPLATES FOR BIBLIOGRAPHIC LABELING

| Type | Feature | Description |
|---|---|---|
| Unigram | <position(0)> | Position of the current token in a token sequence |
| | <f_kanji(0)> | Number of fullwidth kanji in the current token |
| | <f_hiragana(0)> | Number of fullwidth hiragana in the current token |
| | <f_katakana(0)> | Number of fullwidth katakana in the current token |
| | <f_alphabet(0)> | Number of fullwidth alphabets in the current token |
| | <f_digit(0)> | Number of fullwidth digits in the current token |
| | <h_alphabet(0)> | Number of halfwidth alphabets in the current token |
| | <h_digit(0)> | Number of halfwidth digits in the current token |
| | <h_katakana(0)> | Number of halfwidth katakana in the current token |
| | <h_symbol(0)> | Number of symbols in the current token |
| | <#_character(0)> | Number of characters in the current token |
| | <token(0)> | Token itself |
| | <keyword($i$)> | Presence of keywords in the $i$th token<br>($i = -4, -3, -2, -1, 0, 1, 2, 3, 4$) |
| Bigram | <$y$(-1), $y$(0)> | Previous and current labels |

learning parameters such as balancing the degree of fit to default values given by CRF++.

We used the accuracy with which a CRF assigned labels to each bibliography in the test-token sequences as the evaluation metric. A CRF was only regarded as having succeeded in labeling a bibliography when it assigned correct labels to all tokens constituting the bibliography in the test sequence: no more, no less.

The labeling accuracy of each bibliography was

$$\frac{\text{the \# of the sequences having the bibliography successfully labeled}}{\text{the total \# of the sequences having the bibliography}}.$$

We applied five-fold cross validation to the experimental dataset. We also compared the experimental results to those of Okada's method [6] because we used the same experimental data and calculated extraction accuracies on the basis of the same classification of bibliographies as theirs.

### B. Varying Token Granularities

*1) Character Token:* In this setting, each character of a reference string was regarded as a token. The character token was introduced just for reference.

*2) Token by Delimiter Set 1:* For evaluating the effect of token granularity on labeling accuracy, we grouped the delimiters of delimiter set 1 hierarchically, as shown in Table IV. Delimiters in the same group were treated identically during the labeling phase. For example, all kinds of delimiters were treated identically in Set 1-1 while all delimiters were classified into any of four categories (Dsym, Dstr, Dcm, and DSP) in Set 1-4.

*3) Token by Delimiter Set 2:* As with delimiter set 1, we grouped the delimiters of delimiter set 2 hierarchically, as shown in Table V. In the table, "*" means that tokens marked as "*" were identified by using pattern matching and removed before labeling, and "–" means that space characters as delimiters, which appeared only between "month" and "year" (as described in section III-B), were removed before labeling.

### C. Experimental Results

Table VI summarizes the labeling accuracies of major bibliographic elements such as "Author" and whole token sequences of reference strings, "ALL", w.r.t. each delimiter set. As seen in the table, Sets 1-14 and 2-8 were the best performers among the delimiter sets. Although the accuracies of "Day" were considerably worse than the other bibliographic elements irrespective of delimiter settings, it hardly affected the overall accuracy of "ALL" because there were only nine references that had this kind of bibliography among a total of 4,814.

Table VII summarizes the labeling accuracies for character-based and B-I-O tag-generated tokens, as well as Okada's method in addition to those of our best performers, Sets 1-14 and 2-8. If we compare Set 2-8 and Okada's method, Set 2-8 was more accurate in "Author", "Title", "Journal", "Publisher", "Year", and "Other", while Okada's method was more accurate in "Volume", "Day", "Month", and "ALL".

Labeling errors that occurred with the proposed method were often related to inappropriate tokenization, i.e., over- and under-segmentation: more than 40% of mislabeled tokens by Set 2-8 and 60% by Set 1-14 were related to such segmentation errors. We therefore experimented with manually tokenized data. All other conditions were the same as those shown in Table V. The resultant labeling accuracies for tokens acquired by delimiter set 2, shown in Table VIII, indicate that perfect tokenization increased accuracy by more than two percent.

## V. CONCLUSION

We proposed a CRF-based extraction method of bibliographic elements from the reference strings of research papers. The proposed method first segments a reference string into a token sequence and then assigns a bibliographic label to each token in the sequence. Our proposed

Table IV
Delimiter set 1 and its derivative sets

| | D | DS | DE | DZS | DZE | DAND | DED | DN | DPP | DV | DCO | DC | DZCO | DSP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set 1-1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D |
| Set 1-4 | Dsym | Dsym | Dsym | Dsym | Dsym | Dstr | Dstr | Dstr | Dstr | Dstr | Dcm | Dcm | Dcm | DSP |
| Set 1-8 | Dsym | Dsym | Dsym | Dsym | Dsym | DAND | DED | DN | DPP | DV | Dcm | Dcm | Dcm | DSP |
| Set 1-10 | D | Dstart | Dend | Dstart | Dend | DAND | DED | DN | DPP | DV | Dcm | Dcm | Dcm | DSP |
| Set 1-14 | D | DS | DE | DZS | DZE | DAND | DED | DN | DPP | DV | DCO | DC | DZCO | DSP |

Table V
Delimiter set 2 and its derivative sets

| | D | DS | DE | DZS | DZE | DAND | DED | DN | DPP | DV | DCO | DC | DZCO | DSP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set 2-1 | D | D | D | D | D | D | D | D | D | D | * | * | * | – |
| Set 2-2 | Dsym | Dsym | Dsym | Dsym | Dsym | Dstr | Dstr | Dstr | Dstr | Dstr | * | * | * | – |
| Set 2-6 | Dsym | Dsym | Dsym | Dsym | Dsym | DAND | DED | DN | DPP | DV | * | * | * | – |
| Set 2-8 | D | Dstart | Dend | Dstart | Dend | DAND | DED | DN | DPP | DV | * | * | * | – |
| Set 2-14 | D | DS | DE | DZS | DZE | DAND | DED | DN | DPP | DV | DCO | DC | DZCO | DSP |

Table VI
Extraction accuracies of bibliographic elements

| | Author | Title | Journal | Volume | Publisher | Day | Month | Year | Other | ALL |
|---|---|---|---|---|---|---|---|---|---|---|
| Set 1-1 | **0.976** | 0.834 | 0.904 | 0.946 | 0.900 | 0.000 | 0.589 | 0.995 | 0.757 | 0.618 |
| Set 1-4 | **0.976** | 0.849 | 0.932 | 0.976 | 0.901 | **0.500** | 0.796 | **0.998** | 0.764 | 0.704 |
| Set 1-8 | 0.975 | 0.853 | 0.944 | 0.984 | 0.903 | 0.250 | **0.995** | **0.998** | 0.768 | 0.777 |
| Set 1-10 | 0.975 | 0.848 | 0.947 | **0.985** | 0.905 | 0.400 | 0.992 | **0.998** | 0.770 | 0.773 |
| **Set 1-14** | **0.976** | **0.862** | **0.948** | 0.984 | **0.907** | 0.150 | 0.989 | **0.998** | **0.776** | **0.784** |
| Set 2-1 | 0.983 | 0.970 | 0.971 | 0.980 | **0.972** | 0.125 | **0.999** | **0.998** | 0.892 | 0.917 |
| Set 2-2 | **0.987** | 0.976 | **0.975** | 0.981 | 0.969 | 0.125 | **0.999** | **0.998** | 0.895 | 0.925 |
| Set 2-6 | **0.987** | 0.977 | 0.974 | **0.989** | 0.967 | 0.188 | **0.999** | 0.997 | **0.917** | 0.934 |
| **Set 2-8** | **0.987** | **0.980** | 0.974 | 0.988 | 0.967 | 0.188 | **0.999** | **0.998** | 0.916 | **0.935** |
| Set 2-14 | 0.979 | 0.951 | 0.967 | 0.988 | 0.951 | **0.500** | **0.999** | 0.997 | 0.888 | 0.893 |

Table VII
Comparison of extraction accuracies

| | Author | Title | Journal | Volume | Publisher | Day | Month | Year | Other | ALL |
|---|---|---|---|---|---|---|---|---|---|---|
| Character token | 0.445 | 0.490 | 0.481 | 0.872 | 0.201 | 0.000 | 0.967 | 0.966 | 0.345 | 0.368 |
| Set 1-14 | 0.976 | 0.862 | 0.948 | 0.984 | 0.907 | 0.150 | 0.989 | **0.998** | 0.776 | 0.784 |
| Set 2-8 | **0.987** | **0.980** | **0.974** | 0.988 | **0.967** | 0.188 | 0.999 | **0.998** | **0.916** | 0.935 |
| B-I-O | 0.978 | 0.952 | 0.959 | 0.985 | 0.933 | 0.300 | 0.995 | 0.995 | 0.770 | 0.884 |
| Okada's method | 0.970 | 0.965 | 0.970 | **0.995** | 0.840 | **1.000** | **1.000** | 0.995 | 0.905 | **0.976** |

Table VIII
Effect of tokenization quality on extraction accuracies

| | Delimiter set 2 | Perfect tokenization |
|---|---|---|
| Set 2-1 | 0.917 | 0.940 |
| Set 2-2 | 0.925 | 0.947 |
| Set 2-6 | 0.934 | 0.956 |
| Set 2-8 | 0.935 | 0.956 |
| Set 2-14 | 0.893 | 0.939 |

method achieved more than 96% accuracies of extracting major bibliographies from a Japanese academic journal and outperformed Okada's method in the extraction accuracies of more than half of the bibliographies extracted from it. The experimental results also indicated that labeling errors were often related to tokenization errors and that eliminating incorrect tokenization could lead to better accuracy. That is, they indicated more than two percent increases in labeling accuracy of a whole sequence when we used correctly tokenized data. Presently, we are analyzing the labeling errors in more detail so that we can further improve the extraction performance.

## Acknowledgment

This work was supported by JSPS Grant-in-Aid for Scientific Research (B) (23300040), JSPS Grant-in-Aid for Young Scientists (B) (23700119), the Collaborative Research Program of National Institute of Informatics (#49), and Wesco Scientific Promotion Foundation.

## References

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of 18th International Conference on Machine Learning*, 2001, pp. 282–289.

[2] M. Takechi, T. Tokunaga, and Y. Matsumoto, "Chunking-based question type identification for multi-sentence queries," in *Proc. of SIGIR 2007 Workshop on Focused Retrieval*, 2007, pp. 41–48.

[3] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. of EMNLP 2004*, 2004, pp. 230–237.

[4] T. Abekawa, H. Nanba, H. Takamura, and M. Okumura, "Automatic extraction of bibliography with machine learning (in Japanese)," in *IPSJ SIG Technical Report, 2003-FI-72/2003-NL-157*, 2003, pp. 83–90.

[5] F. Peng and A. McCallum, "Accurate information extraction from research papers using conditional random fields," in *HLT-NAACL*, 2004, pp. 329–336.

[6] T. Okada, A. Takasu, and J. Adachi, "Bibliographic component extraction using support vector machines and hidden Markov models," in *Proc. of ECDL 2004*, 2004, pp. 501–512.

[7] M. Ohta, T. Yakushi, and A. Takasu, "Bibliographic element extraction from scanned documents using conditional random fields," in *Proc. of ICDIM 2008*, 2008, pp. 99–104.

[8] M. Ohta, R. Inoue, and A. Takasu, "Empirical evaluation of active sampling for CRF-based analysis of pages," in *Proc. of IEEE IRI 2010*, 2010, pp. 13–18.

[9] E. F. T. K. Sang and S. Buchholz, "Introduction to the conll-2000 shared task: Chunking," in *Proc. of CoNLL-2000 and LLL-2000*, 2000, pp. 127–132.

[10] T. Kudo, "CRF++: Yet another CRF toolkit," http://crfpp.sourceforge.net/.