BibPro: A Citation Parser Based on Sequence Alignment Techniques

Chien-Chih Chen[1], Kai-Hsiang Yang[2], Hung-Yu Kao[3], Jan-Ming Ho[4]
[1,2,4]*Academia Sinica,* [3]*National Cheng Kung University*
rocky@iis.sinica.edu.tw[1], *khyang@iis.sinica.edu.tw*[2], *hykao@mail.ncku.edu.tw*[3],
hoho@iis.sinica.edu.tw[4]

Abstract

The dramatic increase in the number of academic publications has led to a growing demand for efficient organization of the resources to meet researchers' specific needs. As a result, a number of network services have compiled databases from the public resources scattered over the Internet. However, publications in different conferences and journals follow different citation formats, so the problem of accurately extracting metadata from a publication string has also attracted a great deal of attention in recent years. In this paper, we extend our previous work to propose a new tool called BibPro for extracting metadata from citation strings by using a gene sequence alignment tool. The main enhancement of BibPro to our previously tool is that BibPro does not need knowledge databases (e.g., an author name database) to generate feature indices for citation strings. Instead, only the order of punctuation marks in a citation string is used to represent its format. Second, BibPro employs the Basic Local Alignment Search Tool (BLAST) to find the most similar citation formats in database and then uses the Needleman-Wunsch algorithm to choose the best-fit citation format as the extraction template. Our experimental results show that, in terms of precision and recall, BibPro outperforms other existent systems (e.g., INFOMAP and ParaCite), and BibPro can scale well.

1. Introduction

Parsing citation information is essential for integrating bibliographical information published on the Internet, and many related applications, such as field-based searching, academic searching and analysis, and citation analysis [5]. However, it is difficult to design a system to automatically parse citation strings scattered over the Internet because, in addition to the problem of technical typing errors, there are a lot of different citation styles/formats. A citation string usually contains many fields (such as fields of author, title, publication information) arranged in many different formats depending on the type and venues to publish (e.g., for books, journals, conference papers, or technical reports). Hence, it is still challenging to design an automatic system for extracting metadata from citation strings.

Numerous studies on extracting metadata from citation strings are proposed in recent years [1-11, 15]. Those approaches can be classified into three categories: learning-based, template-based and rule-based approaches. Learning-based methods utilize machine learning techniques (e.g.,. the Hidden Markov Model (HMM) [7, 8], Support Vector Machines (SVM) [6], and Conditional Random Fields (CRF) [5]). Among them, CRF achieves the best performance with an overall word accuracy of 95.37% on the Cora reference dataset [5, 16]. Second, template-based methods work by using several template databases with various styles of citation templates (e.g., ParaCite [15] and INFOMAP [11]). ParaCite has been integrated with the EPrints.org software, and links between it with CiteBase, RefLink, and ISI Web of Science are currently considered. INFOMAP is a hierarchical template-based reference metadata extraction method with an overall average accuracy level of 92.39% for the six major citation styles detailed in [11]. Rule-based methods are widely used in real-world applications. For example, CiteSeer [1-4] is a well-known search engine and digital library that uses heuristics to extract certain subfields. It identifies titles and author names in citations with roughly 80% accuracy and page numbers with roughly 40% accuracy [1].

In our previous work [10], we proposed a template-based citation parser that achieved approximately 80% of parsing precision, but it has a number of drawbacks. First, the template construction in our previous work relies on an author name database to identify possible author names, so the quality of the database greatly affects the parsing accuracy, and a high quality author

name database is never easy to obtain in practice. Second, several heuristic rules are applied in our previous work to transform training citation strings into related templates, but these rules only work for several special cases. It causes problems when extracting metadata from a citation string that does not follow those special cases. Third, during the matching process in our previous work, there has high probability to mismatch a wrong template to a citation string because there are several templates having the same similarity score, and no other information could be used to distinguish them. We call this the "template conflict" problem. Generally, the larger the template database, the more serious the problem is.

In this paper, we propose a new citation parser, called BibPro, which retains the advantages of our previous work (e.g., using protein sequences to represent citations and applying the Basic Local Alignment Search Tool (BLAST) to find similar templates), and resolves the weaknesses. Instead of relying on an author name database and heuristic rules, BibPro uses the order of punctuation marks in a citation string as a feature to represent the string's citation format. Furthermore, to find out the template with the highest similarity score, we use the Needleman-Wunsch algorithm [14] in conjunction with BLAST to extract metadata from citation strings and align the features (a protein sequence) with the templates in our template database. Based on these two modifications, BibPro does not need any heuristics, and thus overcomes the template conflict problem. BibPro can effectively and systematically extract the fields of author, title, journal, volume, number (issue), month, year, and page information from citations of different formats.

2. BibPro: Citation Parser

2.1 Basic Ideas

Our system is based on the following two ideas. First, a protein sequence is used to represent a citation string. We split a citation string into several tokens and use an amino acid symbol to represent each token. Figure 1 shows an example of a citation string transformed into a protein sequence "AAADTTTT DLLLLDYRPHS". Second, when transforming a citation string to a protein sequence, only the order of the punctuation marks and reserved words of a citation string are transformed. Redundant information is then filtered out to simplify the problem and accelerate the parsing process. Here, a protein sequence is designed to capture some features of citation strings, and BibPro transform many real citation strings (training set in our system) and their styles to the protein sequences; here, we call these "citation templates" in the following content. When parsing a new citation, BibPro invokes a well developed protein sequence matching program, called BLAST, to search a suitable template and then partitions and extracts the desired metadata (fields) from the citation string.

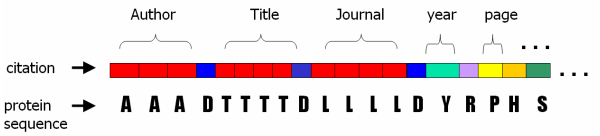

Figure 1. A Transformation from a citation string to a protein sequence

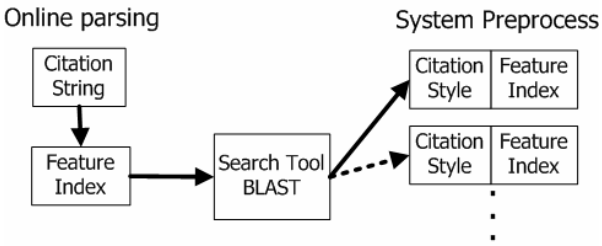

Figure 2. System preprocess and online parsing

Based on these two concepts, BibPro consists of two phases: a system preprocessing phase and an online parsing phase. The goal of the first phase is to generate feature indices for all training citation styles in advance, so that BLAST can find out a suitable citation styles for any given citation string in the second phase (see Figure 2.) During the online parsing phase, BibPro uses BLAST to find a citation style with a feature index similar to that of the citation string. BibPro then is able to extract metadata from this citation string.

2.2. System Architecture

The BibPro system comprises two basic systems: a template generating system and a parsing system, as shown in Figure 3. In the template generating system, we developed programs to retrieve BibTeX files from the Web, and since BibTeX format is field-based, we can easily parse them to get the correct metadata for a citation string. After that, we then use the title field as a search query to search for a citation in CiteSeer or another search engine, e.g., Google. By using this method, we can obtain a lot of citation strings and corresponding metadata. However, the collected data for a citation string may be inconsistent with its metadata. Moreover, our token-based form translation may encounter problems if different fields share the same token. For above reasons, we designed a template filter to ensure that a template is consistent with its citation string. The template filter uses some simple rules (e.g., the author, title and journal fields can not

appear more than once in a citation string). After this process, BibPro can construct a large number of citation templates, and each of which includes a citation style and a feature index.

Once the template database has been compiled, BibPro can provide the citation parsing service on-the-fly. In the parsing system, when a queried citation string is inputted, BibPro transforms it into a protein sequence, and uses BLAST to search candidate citation templates from template database by matching their feature indices. BibPro then uses the Needleman-Wunsch algorithm to calculate the similarity between candidate citation templates and the queried citation string, and the most fit citation template is chosen. According to the final citation template, BibPro can extract metadata from the queried citation string.

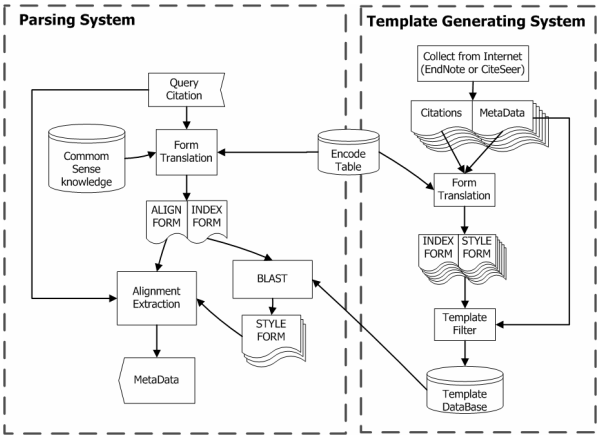

Figure 3. System architecture of BibPro

2.3. Form Translation

In order to use the BLAST to match similar citation templates, we need to transform incompatible citation strings into compatible protein sequences. Therefore, we need to consider the following questions:

- How many symbols can be used in a protein sequence?
- How many fields should be extracted from a citation string?
- How do we transform a citation string into a protein sequence and retain its citation style information?

Having considered the above questions, we created an encoding table, as shown in Table1, to define the relationships between the tokens in a citation string and the symbols in protein sequences.

Table 1. Encoding Table

A: Author T: Title L: Journal F: Volume value W: Issue value	N: numeral Q: @ # $ % ^ & * + = \ | ~ _ / ! ? 。 I: ([{ < 「
H: Page value X: noise (unrecognized token) M: Month Y: Year (number: 1900-2010) S: Issue key. e.g. “no”, “No” P: Page key. e.g. “pp”, “page” V: Volume key. e.g. “Vol”, “vo”	K:)] } > 」 D: . G: " “ ” R: , C: - : E: ' ` Z: ; B: blank (use one “B” to replace continuous “X”)

The design of the encoding table is based on the following observations:

- BLAST can only process sequences with 23 different symbols, so we use these 23 symbols to represent different fields, and use field separators to keep the citation style information in sequence.
- The most common fields in citation strings include: author, title, journal, volume, number, page, issue, month and year. We focus on extracting these fields from citation strings and assign a symbol to represent each field.
- The most common reserved words in citation strings include: "vo", "vol", "no", "NO", "pp", and "page". Since these words are also used to separate fields, we use a symbol to represent each kind of reserved words.
- The punctuation marks usually are used to separate fields, including: " , ", " . ", " ; ", " : ", " " " and " ' ". We also assign a symbol to represent each punctuation mark.
- Brackets and parentheses are synonymous in citation strings, so we use one symbol to represent both.
- Several kinds of punctuation marks appear in the title field, such as: " - ", " ! ", " ? ". However, we only use one symbol to represent all of them because these marks are useless.

Figures 4 and 5 show examples of citation strings transformed into protein sequences. Figure 4 shows that when the correct answer of a citation string's partitions is known, we can correctly label each token. We use the "RESULT FORM" to represent the correct encoded protein sequence. If we can correctly transform a citation string to its RESULT FORM, it is easy to extract metadata from the citation strings. However, it is impossible to know the correct answer when online parsing a citation string. As shown in Figure 5, we can only label each token based on its content due to no other information we could have. If some unrecognized tokens are found, we replace them with an "X". We call this protein sequence the "BASE FORM". Thus, the goal of parsing process is to transform a citation string from its BASE FORM to its RESULT FORM.

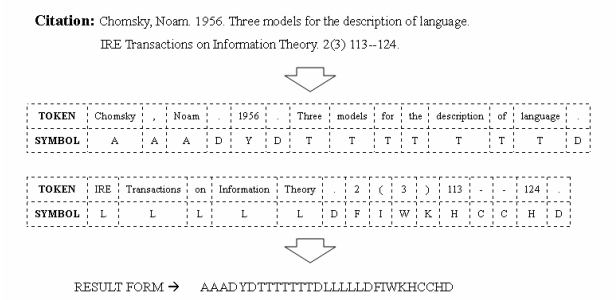

Figure 4. RESULT FORM of a citation string

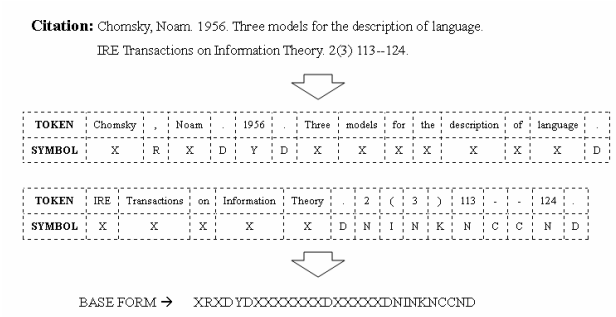

Figure 5. BASE FORM of a citation string

To transform a citation string from its BASE FORM into its RESULT FORM, we need to know its citation style. For this reason, we define several forms of a protein sequence for our mining process:

- STYLE FORM: To store citation style information. Although RESULT FORM can represent style information for specific citation strings, but it has a lot of excess information, such as the length of author, title, and journal fields. Thus we condense the redundant information in the RESULT FORM by using one of each symbol to represent the above fields. This sequence, called the "STYLE FORM", is used to represent a citation style.
- INDEX FORM: To recognize the style of a citation string, we use the order of punctuation marks in citation strings as the feature index. By removing all other unrecognized tokens. The INDEX FORM is the protein sequence that BLAST will try to match with similar sequences in the template database, so it is like an index used in a database.
- ALIGN FORM: Many fields of the citation like the author, title, and journal fields may contain punctuation marks. To extract these fields correctly, we have to remove punctuation marks inside the field. By some common sense knowledge, such as a dot always follow a name initial, we mark and group author and journal field tokens in citation strings during the online parsing phase. The processed sequence, called the "ALIGN FORM", is used to represent original citation string.

Table 2 shows an example of each form.

Table2. Example of each form

BASE FORM	XRXDYDXXXXXXXDXXXXXDNINKNCCND
RESULT FORM	AAADYDTTTTTTTDLLLLLDFIWKHCCHD
STYLE FORM	ADYDTDLDFIWKHCCHD
INDEX FORM	RDYDDDNINKNCCND
ALIGN FORM	XXXDYDXXXXXXXDXXXXXDNINKNCCND

When parsing a citation string, BibPro use the Needleman-Wunsch algorithm to perform global alignment between the STYLE FORM and the ALIGN FORM. With the alignment, BibPro is able get the RESULT FORM from the ALIGN FORM by adding "A" (author), "L" (journal), and "T" (title) in the correct positions and by changing "N" to its corresponding amino acid (e.g., an amino acid "N" may become F [volume], "W" [issue] or "H" [page]) as shown in figure 6. After that, by checking the original citation string and the RESULT FORM, BibPro can extract all the metadata correctly.

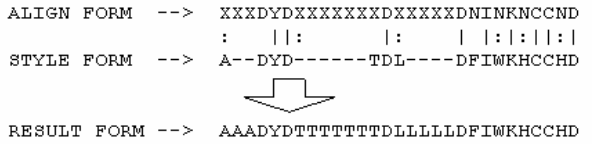

Figure 6. Align STYLE FORM and ALIGN FORM to get RESULT FORM

3. Experiment

3.1. Datasets

We chose two datasets for our experiments. The first dataset used in [12] comprises six citation styles, namely, JMIS, ACM, IEEE, APA, MISQ, and ISR, and includes total 160,000 citation strings. To simulate experiments in [11], we randomly selected 10,000 strings as the training set to generate the template database and another 10,000 citation strings for testing. We refer this dataset as D1.

The second dataset was obtained from CiteSeer and 6,500 citation strings are gathered. We developed two programs: one to retrieve the BibTeX files from each citation string on the Internet; and the other for choosing the title field to search the citations in CiteSeer so that we could compile the citation strings and their corresponding metadata. We used 2,500 citation strings for training and 4,000 for testing. We refer this dataset as D2.

The citation strings in D1 are more regular than those in D2 because they were generated from existing data. Moreover, the citation styles only differ in the order of the fields and the separators of the fields. In other words, D1 has smaller variations in the citation

formats. We therefore collected real data from the Internet to generate the D2 dataset, which is more varied and fits real-world applications better.

3.2. Performance Measurements

We use different performance measurements for the datasets in our experiments. The first measurement, which is defined in [11]:

$$\text{Accuracy} = \frac{\text{Number of correctly extracted fields}}{\text{Total number of fields}}$$

This accuracy measurement, called EVAL1, is used to evaluate the system's performance on the D1 dataset.

The second measurement is used for the D2 dataset. For this dataset, the metadata in BibTeX that we collected from the Internet should be consistent with the metadata of the citation string. Unfortunately, some of the BibTeX metadata from the Internet does not fit the corresponding citation string. To resolve this problem, we developed the following measurement to determine whether the data is correctly parsed.

$$\text{Field Precision} = \frac{\#[\text{Token}_{\text{parsed field}} \cap \text{Token}_{\text{BibTex field}}]}{\#[\text{Token}_{\text{query citation}} \cap \text{Token}_{\text{BibTex}}]}$$

where $\text{Token}_{\text{parsedfield}}$ denotes tokens that appear in the parsed subfield; $\text{Token}_{\text{query citation}}$ denotes tokens that appear in the query citation string; $\text{Token}_{\text{BibTex field}}$ denotes tokens that appear in a specific subfield in the BibTeX file; and $\text{Token}_{\text{BibTex}}$ denotes all tokens that appear in the BibTeX file.

The denominator represents the number of the tokens in both the citation string and the BibTeX file, while numerator represents the number of correctly parsed tokens. We use this measurement, called EVAL2, to compare BibPro with ParaCite.

Using these two measurements, we can compare BibPro with other systems and derive more reliable experiment results.

3.3. Experimental Results

3.3.1. Comparison with INFOMAP. The first experiment compares BibPro with INFOMAP [11]. We used EVAL1 on the D1 dataset; the results are shown in Table 3. It is easy to observe that BibPro outperforms INFOMAP with an overall average accuracy for the six styles of 97.68% versus 92.39% for INFOMAP. Furthermore, in all fields, except the journal field, BibPro achieves a much higher average accuracy level than INFOMAP. More specifically, BibPro is at least 5% more accurate in the author, title, issue and page fields. Similarly, of the six different citation styles mentioned earlier, BibPro excels in all styles except the MISQ style. The results show that BibPro achieves a better performance than INFOMAP. Furthermore, it is reliable when the dataset is regular and clean.

Table 3. Extraction results of BibPro and INFOMAP on D1 using EVAL1

	Style	author	title	journal	volume	issue	year	page	overall avg
Bib Pro	APA	99.67%	96.38%	97.06%	98.99%	98.12%	99.42%	98.71%	98.33%
	IEEE	98.72%	98.12%	99.12%	99.30%	98.39%	99.40%	98.40%	98.78%
	ACM	97.14%	95.01%	93.93%	97.19%	97.03%	98.88%	97.92%	96.73%
	ISR	99.48%	96.17%	96.96%	99.15%	98.39%	99.35%	98.55%	98.29%
	MISQ	98.59%	97.99%	98.98%	99.41%	98.61%	99.54%	98.83%	98.85%
	JMIS	91.95%	87.90%	90.46%	99.23%	98.03%	99.46%	98.76%	95.11%
	Avg.	97.59%	95.26%	96.09%	98.88%	98.09%	99.34%	98.53%	97.68%
INFO MAP	APA	92.32%	71.80%	94.33%	97.39%	84.92%	96.48%	95.09%	90.33%
	IEEE	94.17%	89.05%	92.07%	95.45%	84.49%	97.18%	89.81%	91.75%
	ACM	88.36%	91.10%	99.41%	80.28%	87.73%	96.47%	83.95%	89.61%
	ISR	91.93%	78.33%	95.32%	95.28%	87.00%	96.34%	90.61%	90.69%
	MISQ	97.73%	97.92%	100%	99.99%	99.98%	99.94%	99.64%	99.31%
	JMIS	76.55%	72.57%	99.99%	99.98%	99.97%	99.93%	99.69%	92.67%
	Avg.	90.18%	83.46%	96.85%	94.73%	90.68%	97.72%	93.13%	92.39%

3.3.2. Comparison with ParaCite. In this experiment, we compared BibPro with ParaCite [15] using the EVAL2 as the performance measurement on the D2 dataset. The results are detailed in Table 4. Since the source code for ParaCite is available on the Internet, we can use the D2 dataset, which was compiled by our automatic programs to compare ParaCite's performance with that of BibPro. Because ParaCite does not automatically build templates, we use ParaCite's default template database to test the D2 dataset, which contains about 4,000 records. Moreover, because ParaCite can only extract one author name per citation string, its accuracy in the author field is much lower than that of BibPro. From Table 4, we can observe that, in terms of accuracy, BibPro outperforms the ParaCite system by more than 20% in all fields, except the title field, and by as much as 90% in the page field. BibPro achieves a better performance than ParaCite because the D2 dataset consists of real data, which is more complex than regular datasets. However, comparing the accuracy level of the different fields in BibPro, it is interesting to note that the average accuracy for the title and journal fields is consistently lower than it is for other fields. This is probably due to the frequent variability (the variability in punctuation e.g., "-", ".", and "?") in the title and journal fields.

Table 4. Extraction results of ParaCite and BibPro on D3 using EVAL3

	Author	Title	Journal	Volume	Page	Issue	Month	Year
Bib Pro	93.11%	73.31%	54.23%	82.79%	95.08%	84.63%	88.99%	96.47%
Para Cite	24.02%	72.77%	29.65%		4.67%	24.57%		77.02%

4. Conclusion and Future Work

Parsing citations is still a challenging problem due to the diverse nature of citation formats. In this paper, we proposed a template-based citation parsing system called "BibPro", which extends our previous work by using the order of punctuation marks in a citation string to represent its format. When online parsing a citation string, BibPro transforms the citation string into a protein sequence and apply two sequence alignment techniques, BLAST and the Needleman-Wunsch algorithm, to find out the most similar template for exaction metadata from the citation. According to our experiments, BibPro performs very well and is scalable.

There are still several challenges when applying the BibPro into real world applications. One challenge is to obtain an accurate large-scale training dataset with all kinds of citation formats. The training dataset collected from the Web always contains a lot of errors, such as missing values, spelling errors, inconsistent abbreviations, and extraneous tokens [9]. Another challenge is that many publication formats include a lot of fields, and it is difficult to extract all the fields for all citation strings. Hence, we only concentrate on the most common information (fields) for all publication formats in this paper.

In the future, we focus on designing an automatic system to extract all publication information from researchers' publication lists by integrating the BibPro with our previous work [17]. We believe that more research in these areas would definitely be worthwhile.

5. References

[1] Giles, C. L., Bollacker, K. D., and Lawrence, S. "CiteSeer: an automatic citation indexing system," *Digital Libraries 98* Pittsburgh PA USA, 1998.

[2] Bollacker, K. D., Lawrence, S., and Giles, C. L., "CiteSeer: an autonous Web agent for automatic retrieval and identification of interesting publications," 1998.

[3] Lawrence, S., Giles, C. L., and Bollacker, K. D., "Autonomous citation matching," 1999.

[4] Lawrence, S., Giles, C. L., and Bollacker, K. D. "Digital Libraries and Autonomous Citation Indexing." *IEEE Computer*. Vol 32, 1999, pp. 67-71.

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

[6] Hui Han, Giles, C.L., Manavoglu, E., Hongyuan Zha, Zhenyue Zhang, Fox, E.A. "Automatic document metadata extraction using support vector machines," *JCDL*, 2003.

[7] K. Seymore, A. McCallum, R. Rosenfeld, "Learning hiddenMarkov model structure for information extraction," *AAAI-99*, 1999, pp. 37-42.

[8] Takasu, A. "Bibliographic attribute extraction from erroneous references based on a statistical model," *JCDL*, 2003, pp. 49-60.

[9] Agichtein, E. and Ganti, V., "Mining reference tables for automatic text segmentation", *KDD'04*, 2004, pp. 20-29.

[10] I-Ane Huang, Jan-Ming Ho, Hung-Yu Kao, and Shian-Hua Lin, "Extracting citation metadata from online publication lists using BLAST," *In PAKDD*, 2004.

[11] Min-Yuh Day et al. "Reference metadata extraction using a hierarchical knowledge representation framework." *Decision Support Systems*, 2006.

[12] S. F. Altschul, W. Gish, W. Miller, E. Myers and D. Lipman. "A basic local alignment search tool." *J. Mol. Biol.*, 215, 1990, pp. 403-410.

[13] http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/similarity.html

[14] Needleman, S. B. and Wunsch, C. D. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *J. Mol. Biol.*, 48, 1970.

[15] http://paracite.eprints.org/

[16] http://www.cs.umass.edu/~mccallum/code-data.htm

[17] K.-H. Yang, J.-M. Chung and J.-M. Ho, "PLF: A Publication List Web Page Finder for Researchers", in the 2007 IEEE/WIC/ACM International Conference on Web Intelligence, 2007.