Bilingual PRESRI
Integration of Multiple Research Paper Databases

Hidetsugu Nanba[1], Takeshi Abekawa[2], Manabu Okumura[3], Suguru Saito[3]

[1] School of Information Sciences, Hiroshima City University
nanba@its.hiroshima-cu.ac.jp
[2] Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
abekawa@lr.pi.titech.ac.jp
[3] Precision and Intelligence Laboratory, Tokyo Institute of Technology
{oku, suguru}@pi.titech.ac.jp

Abstract

Collecting all the papers in a research field is a first step towards an exhaustive survey. A number of research paper databases are available for searching papers. However, searchers are compelled to repeat the same search operation for each database if there are multiple databases for a research field. To improve such inefficient searching, we have developed PRESRI, which can construct an exhaustive database by integrating multiple research paper databases. First, we collect Postscript and PDF files on the WWW, and construct a database (‘WEB-DB’) by extracting bibliographic information from the files. Second, we construct an exhaustive database by integrating WEB-DB with other databases. As a key technique for constructing an exhaustive database, we propose a method for extracting bibliographic information from Postscript and PDF files based on a SVM. To investigate the effectiveness of our method, we conducted an examination. We found that our method is useful for both Japanese and English. In this paper, we also focus on the presentation of search results, which is an important factor in constructing an efficient survey environment. We have developed a system that makes it possible to understand the relationships between papers intuitively based on citation information.

1. Introduction

Collecting all the papers, written in various languages, in a research field is a first step towards an exhaustive survey. A number of research paper databases, which are provided by libraries, publishing companies and academic societies, are available for searching papers. However, searchers are compelled to repeat the same search operation for each database when there are multiple databases for a research field. To improve such inefficient searching, we aim to develop a system called PRESRI (**P**aper **RE**trieval **S**ystem using **R**eference **I**nformation), which can construct an exhaustive database by integrating multiple research paper databases.

First, we collect Postscript and PDF files on the World Wide Web, and construct a database (we call the database ‘WEB-DB’) by extracting bibliographic information from the files. Second, we construct an exhaustive database by integrating WEB-DB with several other databases. Several methods for extracting bibliographic information from Postscript and PDF files have been proposed (Bergmark *et al.* 2001; Ding *et al.* 1999; Seymore *et al.* 1999; Borkar *et al.* 2001; Connan *et al.* 2000; Geng 2003; McCallum *et al.* 1999; Takasu 2003), but they focus mainly on English. To construct an exhaustive database covering various languages, we propose a new extraction method for English and Japanese, which can easily be expanded to many other languages.

In this paper, we also focus on the presentation of search results. Traditional search engines show search results as a list, even if there are more papers than can be read. To grasp the outline of search results quickly, we have developed an efficient presentation system, which makes it possible to understand the relationships between papers intuitively, using citation relationships, which show explicit relations between papers. In the field of citation analysis, citation relationships have been used for classifying papers, evaluating the importance of papers or journals, and analysing the relationships between research fields. However, most analyses treat all citations equally, although there are actually several reasons for citations. In our system, citation relationships are classified automatically according to the reasons for them (we call them ‘citation types’) in advance of search operations.

When a search operation is conducted, citation relationships with their citation types are shown visually as a search result.

In the remainder of the paper, Section 2 describes some essential points for constructing an exhaustive and efficient survey environment. Section 3 describes a procedure for constructing PRESRI and a system configuration. Section 4 describes our method for extracting bibliographic information from headers and lists of references from Postscript and PDF files. We also report experimental results. Section 5 shows system behaviour with snapshots. Section 6 presents conclusions and prospects for further research.

2. Essential Points for Constructing PRESRI

In this section, we describe some essential points and fundamental ideas for constructing PRESRI. The detailed procedure for constructing PRESRI will be explained in Section 3.

2.1. Constructing an Exhaustive Database

There are several free databases related to WEB-DB. Both CiteSeer (Lawrence *et al.* 1999) and Cora (McCallum *et al.* 1999) are famous as full-text citation indices, and have been constructed automatically. ‘arXiv.org’[1] is another free database including Postscript, PDF and TeX files, which are registered by the researchers themselves. However, these databases deal with English papers only. To construct an exhaustive database, it is necessary to deal with papers written in various languages. As a first step towards constructing a multi-lingual database, we deal with English and Japanese papers in Postscript and PDF format on the Web.

Copyright in databases should also be taken into account when integrating multiple databases. There are many pay databases, such as on-line journals provided by publishing companies or academic societies, and CD-ROMs or DVDs of conference proceedings. Databases created by university libraries may also be pay databases. These databases must be integrated on a local server with other databases, such as WEB-DB, to allow their use by particular people or those at particular places. We therefore aim to integrate local and remote databases on a local server. We will explain the detailed system configuration in Section 3.

2.2. Efficient Presentation of Search Results

Traditional search engines show search results as a list, even if the results are too numerous for all of them to be read. We therefore focus on efficient presentation of search results. We first explain citation information, which is useful in this context. Next, we describe a visual interface for effective presentation of citation information.

2.2.1. Citation Information We will briefly explain our previous work of extracting citation information and its application in supporting surveys (Nanba *et al.* 1999). In a research paper, there are passages where the author of a current paper describes the essence of cited papers and the differences between the current paper and the cited papers (we call these passages ‘citing areas’). We call the information derived from these passages ‘citation information’. With the information from citing areas, we can see the similarities and differences between the current paper and the cited papers. We can also identify the reasons for citation. Citation information makes it possible to understand the stance of a paper among several related papers, or to grasp the outline of the domain. We therefore use citation information for the presentation of search results.

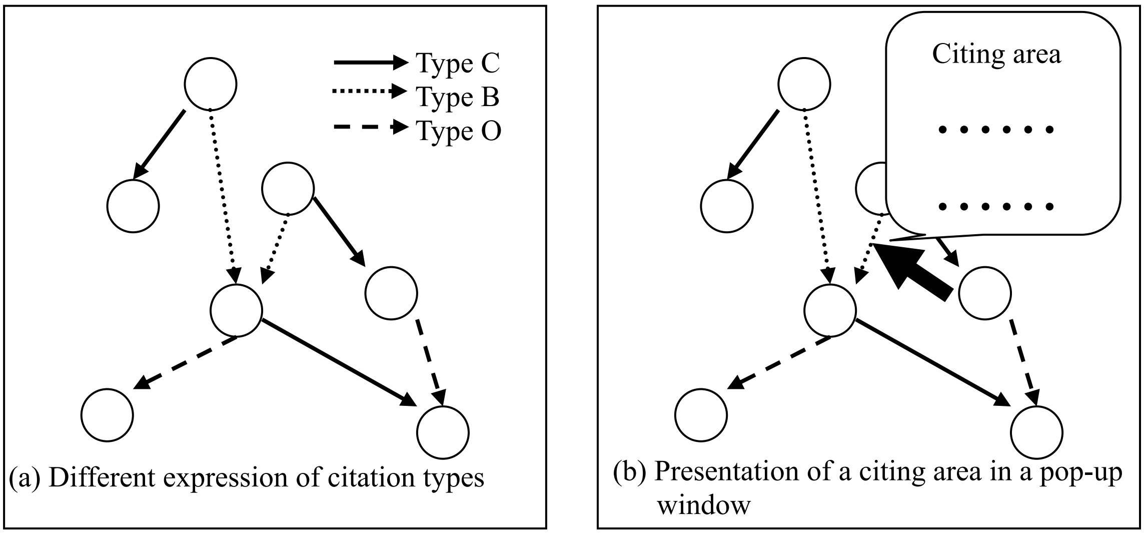
Extraction of citation information consists of two processes: extraction of citing areas and identification of citation types. A citing area is defined as a succession of sentences that have a connection with the sentence that includes the citation in the paragraph. As we believed that such a connection between sentences could be indicated by some cue phrases, we used those cue phrases for citing area extraction. We then proposed a method for identifying the following citation types automatically using several cue phrases.

[1] http://www.arxiv.org

- **Type B**: Citations that base the current work on other researchers' theories or methods.
- **Type C**: Citations that compare with related papers or point out their problems.
- **Type O**: Citations other than types B and C.

In a citing area, if a negative expression like 'However' or 'cannot' appears in the sentence after the sentence containing the citation, the citing area can be considered as type C. Similarly, if an expression like 'we adopt' or 'we use' appears in the sentence including the citation, the citing area can be considered as type B. We prepared a list of such cue phrases and made 160 rules using them for the automatic determination of citation types. Similar citation types were proposed by Teufel *et al.*(Teufel *et al.* 2000, Teufel 2001). They identified the structure of scientific argumentation in articles, including CONTRAST, BASIS and OTHERS, which correspond to types C, B and O, respectively.

2.2.2. Visualization of Search Results with Citation Information In this paper, we attempt to present search results visually in consideration of citation information, as well as showing search results as a list, like CiteSeer and Cora. One method of visualizing search results is to present a graph of citations, i.e. papers and citation relationships are shown as dots and arrows, respectively. There has been much research on visualizing graphs such as citation relationships, a file hierarchy on a computer system, and website maps (Herman *et al.* 2000). In presenting citation information, we must present citation types and citing areas in addition to citation relationships. Here, citation types are expressible as different kinds of arrows in a graph, as shown in Figure 1(a). In the figure, types C, B and O are expressed as continuous, dotted, and dashed lines, respectively. In the citation graph in Figure 1(a), citing areas can be shown in a pop-up window as shown in Figure 1(b), if a user puts a cursor on an arrow. We attempt to implement such a visual interface of search results.



(a) Different expression of citation types

(b) Presentation of a citing area in a pop-up window

Figure 1: Presentation of citation information

3. Construction of PRESRI

3.1. Procedure for Constructing PRESRI

PRESRI is constructed in two stages: (1) extraction and (2) integration of bibliographic information.

In the extraction stage, bibliographic information (titles, authors, affiliations of authors, keywords, and abstracts) and a list of references are extracted from each paper (Postscript or PDF). In some Japanese research papers, bibliographic information is written in both Japanese and English, so we extract both and store them in a database. Details of our extraction method will be described in Section 4. We then identify citation types automatically (Nanba *et al.* 1999).

In the integration stage, extracted bibliographic information, abstracts, and lists of references from the extraction stage are gathered and integrated. A key technique in this stage is to identify duplicate interlingual and intralingual bibliographic information. The procedure for identifying duplicate intralingual bibliographic information is as follows:

1. Eliminate particular marks, such as punctuation and hyphens, from both titles,
2. Compare years of publication,
3. Measure the ratio of string match between titles using DP-matching,
4. Identify bibliographic information if the ratio exceeds a threshold value and the publication years match (If one or both publication years are not extracted in the extraction stage, we consider that the years match).

Almost in the same way as intralingual identification, we can identify interlingual bibliographic information. When a Japanese paper is cited in an English paper, '(in Japanese)' is written after bibliographic information in a list of references. If we find '(in Japanese)' in a citation, it is only necessary to search Japanese bibliographic information with the English title and authors, which are extracted from Japanese papers in the extraction stage.

3.2. System Configuration

The system consists of the several servers and modules. The system configuration is shown in Figure 2.

Servers

- **Local Servers**

The roles of local servers are to provide a search function for users, and to integrate multiple local databases (e.g. CD-ROMs) with remote databases (e.g. WEB-DB) via a Web browser interface. The servers are situated in universities, research organizations, and so on.

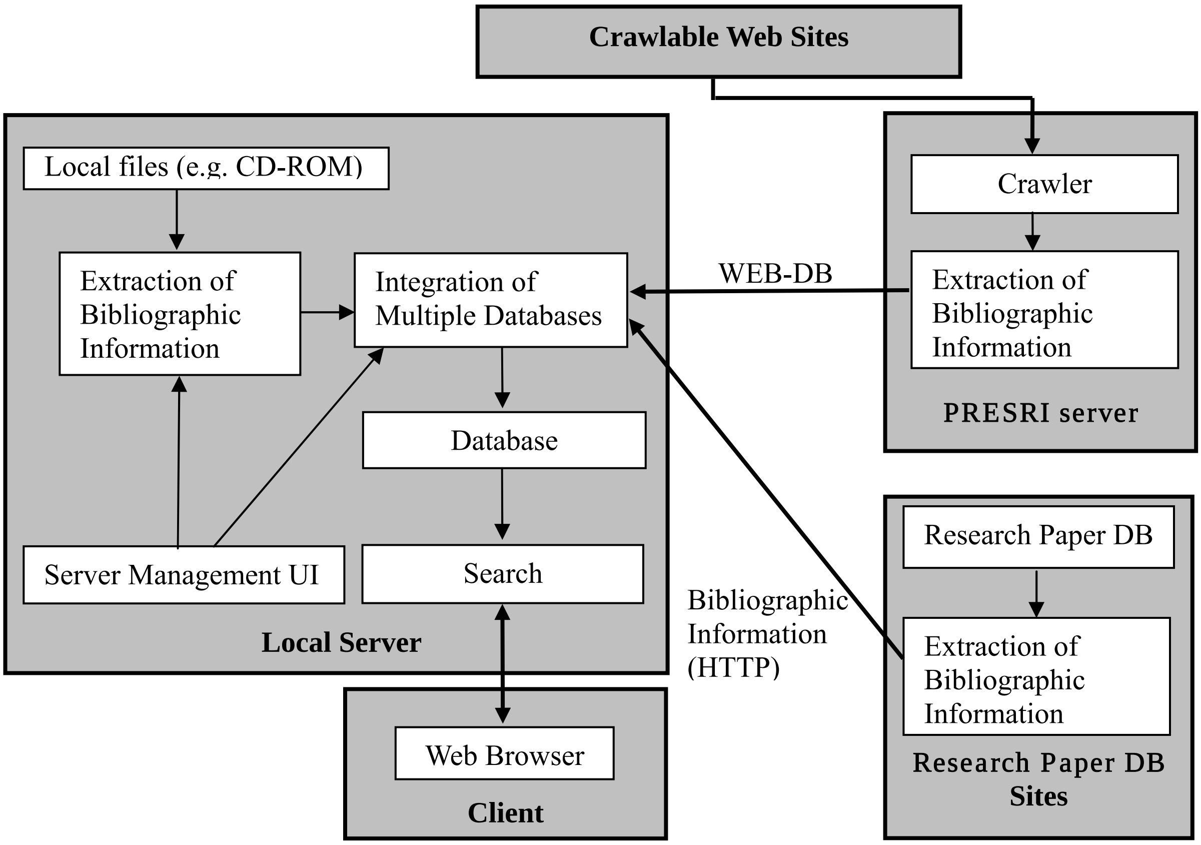

Crawlable Web Sites
Local files (e.g. CD-ROM)
Extraction of Bibliographic Information
Integration of Multiple Databases
Database
Server Management UI
Search
Local Server
Web Browser
Client
WEB-DB
Crawler
Extraction of Bibliographic Information
PRESRI server
Bibliographic Information (HTTP)
Research Paper DB
Extraction of Bibliographic Information
Research Paper DB Sites

Figure 2: System configuration

- **PRESRI Server**

PRESRI server collects Postscript and PDF files on the Web, and extracts bibliographic information from collected data. We call the data WEB-DB.

- **Clients**

If only Web browsers are installed in clients, users can search papers via Web browsers.

- **Research Paper DB Sites**

Research paper DB sites hold one or more databases that robots are not permitted to crawl. If administrators of databases allow specific users to integrate them with other databases, bibliographic information, citation relationships, and citation types are extracted on their servers. They can then be integrated on the specific users' local servers.

- **Crawlable Web Sites**

Crawlable Web sites are those that permit Web crawlers to access (and also collect) Web pages. Typical crawlable Web sites are researchers' private sites, where researchers put their research papers (Postscript or PDF). The sites permit Web crawlers to collect the papers.

Modules

- **Crawler**

A crawler accesses servers within the academic domain ('ac' and 'edu' domains) on the Web, and collects Postscript and PDF files. Currently, we use 'wget[2]' as a crawler.

- **Extraction of Bibliographic Information**

This module operates on the PRESRI server, sites of research paper DBs, and local servers. The module converts Postscript or PDF files into XML files, and extracts bibliographic information from their headers and lists of references using linguistic information and information about text appearance (e.g. font size, bold, italic). Extracted data are saved in an appointed directory. Details of this module will be explained in Section 4.

- **Integration of Multiple Databases**

This module integrates multiple databases and registers them for a local server in the following three steps. First, the module collects bibliographic information from one or more sources that have been registered in advance for the local server via the server management UI. Second, the module identifies duplicate interlingual and intralingual bibliographic information among different databases, and integrates it into one bibliographic database. Third, the database is registered for a local server.

- **Server Management UI**

Administrators of local servers can register new local and remote databases, and integrate them with others via the server management UI.

- **Database**

This module provides a function for setting up data sources and for controlling the two modules 'extraction of bibliographic information' and 'integration of multiple databases'. 'Extraction of bibliographic information' can be operated at fixed intervals for each database to follow periodical updates of the data source.

- **Search**

This module provides functions for searching papers and for displaying citation relationships.

[2] http://www.gnu.org/software/wget/wget.html

4. Automatic Extraction of Bibliographic Information

In this section, we propose two methods for extracting bibliographic information from both headers and lists of references in Postscript and PDF files. We first describe a procedure for extracting bibliographic information. Then, we describe our methods in detail. We conducted two studies and we report the experimental results.

4.1. Procedure for Extracting Bibliographic Information

The procedure for extracting bibliographic information is shown in Figure 3.

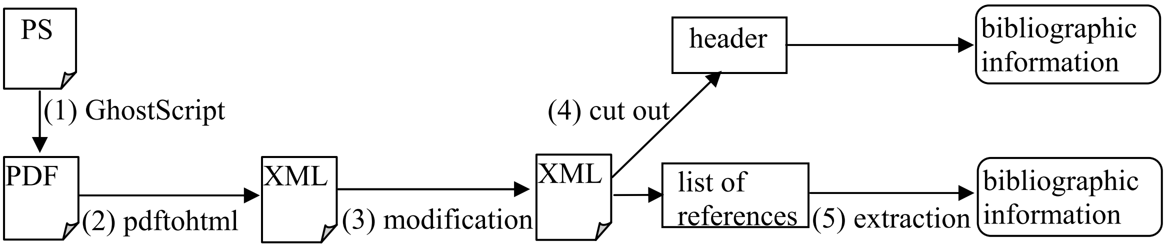


Figure 3: Procedure for extracting bibliographic information

(1) Postscript files are converted to PDF files using GhostScript.
(2) PDF files are converted to XML files using pdftohtml[3]. The output XML files include plain text with appearance information for each text box, such as font size, font style (e.g. bold, italic), and coordinate values are calculated from the upper left corner of the page.
(3) Some surface modifications are conducted. For example, ligatures (e.g. 'fl' or 'ffi') are divided into single characters in this stage.
(4) Headers and lists of references are cut out. Here, a header is defined to be the beginning of the paper up to the end of the first page or the first section.
(5) Bibliographic information is extracted from headers and lists of references.

In the remainder of this section, we describe the extraction from headers and then explain the extraction from lists of references.

4.2. Extraction of Bibliographic Information from Headers

4.2.1. Related Works Several methods have been proposed for extracting components of papers, such as titles, authors, keywords and other information, from headers. Bergmark *et al.* and Ding *et al.* devised several rules to extract titles and authors from headers, taking account of information from text appearance (Bergmark *et al.* 2001; Ding *et al.* 1999). However, simple rule sets could not extract titles and authors from various styles of papers. Instead of making rules manually, Seymore *et al.* applied Hidden Markov Models (HMM) to the extraction of bibliographic information (Seymore *et al.* 1999). They regarded each state as a class, such as 'title', 'author', or 'affiliation', and treated word sequences as observation symbols. Each state emits words from a class-specific unigram distribution. However, their HMM-based model could not deal with both linguistic information and information from text appearance at the same time.

We use linguistic information and information from text appearance for extraction of bibliographic information from headers. Instead of making rules manually, we apply the Support Vector Machine (SVM) for the extraction, which makes it possible to use a variety of features effectively (Vapnik, 1998).

[3] http://pdftohtml.sourceforge.net/

4.2.2. Experiments To investigate the effectiveness of our method, we conducted an experiment.

Data Sets

The data sets used in our experiment are shown in Table 1. Most of them are in the field of computer science.

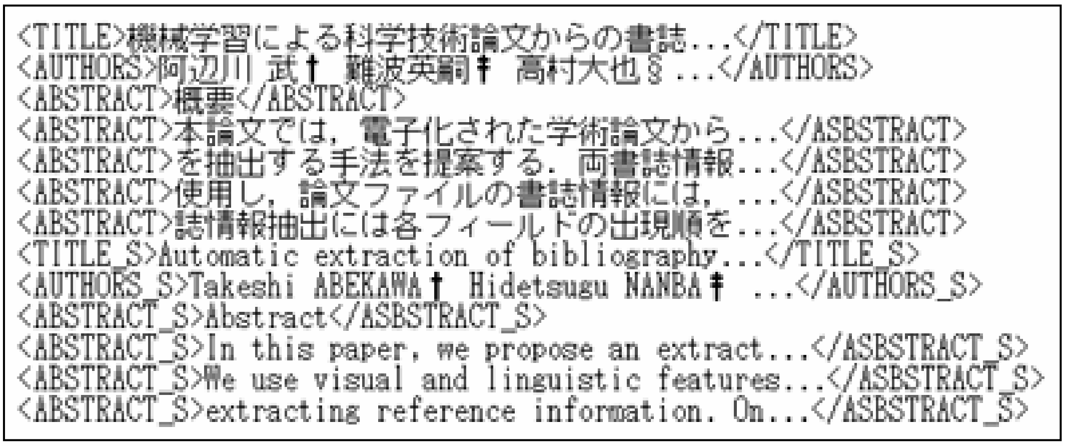
Data sets	Number of files	Headers		References	
		English	Japanese	English	Japanese
ACL 2003	65	65	0	150	0
COLING 2003	140	140	0	150	0
IEICE 2003	150	8	142	223	147
IPSJ 2003	177	1	176	150	236
JSAI 2003	208	5	203	152	244
IPSJ SIG-NL	98	2	96	150	232
Data on the WWW	107	73	34	147	96
Total	945	294	651	1122	955

Table 1: Data sets used in an experiment for extracting bibliographic information from headers

For these data sets, we added tags as shown in Table 2 to each line in the XML files (the output files of step 3 in Figure 3). In Table 2, 'AFFILIATION' includes authors' affiliations, phone numbers, addresses and URLs, and excludes e-mail addresses. If headers include multiple bibliographic information written in different languages, we first annotate using a set of 'Main language' tags, then annotate using a set of 'Sub-language' tags. Part of the tagged data is shown in Figure 4. The latter part of each line is omitted.

Field name	**Main language**	**Sub language**
Title	TITLE	TITLE_S
Author	AUTHORS	AUTHORS_S
Affiliation	AFFILIATION	AFFILIATION_S
Abstract	ABSTRACT	ABSTRACT_S
Keyword	KEYWORD	KEYWORD_S
Email	EMAIL	
Other	OTHER	

Table 2: Tag sets used in machine learning



```
<TITLE>機械学習による科学技術論文からの書誌...</TITLE>
<AUTHORS>阿辺川 武† 難波英嗣‡ 高村大也§ ...</AUTHORS>
<ABSTRACT>概要</ABSTRACT>
<ABSTRACT>本論文では，電子化された学術論文から...</ASBSTRACT>
<ABSTRACT>を抽出する手法を提案する．両書誌情報...</ASBSTRACT>
<ABSTRACT>使用し，論文ファイルの書誌情報には，...</ASBSTRACT>
<ABSTRACT>誌情報抽出には各フィールドの出現順を...</ASBSTRACT>
<TITLE_S>Automatic extraction of bibliography...</TITLE_S>
<AUTHORS_S>Takeshi ABEKAWA† Hidetsugu NANBA‡ ...</AUTHORS_S>
<ABSTRACT_S>Abstract</ASBSTRACT_S>
<ABSTRACT_S>In this paper, we propose an extract...</ASBSTRACT_S>
<ABSTRACT_S>We use visual and linguistic features...</ASBSTRACT_S>
<ABSTRACT_S>extracting reference information. On...</ASBSTRACT_S>
```

Figure 4: Part of tagged data

Feature Sets for Machine Learning

The feature sets used for the SVM are as follows,

- **Information from text appearance:**
  - ♦ Distance from the top of the page (divided by the page height, for normalization)
  - ♦ Distance from the left edge of the page (divided by the page width, for normalization)
  - ♦ Width of a box enclosing a line (divided by the page width, for normalization)
  - ♦ Five graded font sizes. For example, the third largest font is expressed as (0,0,1,0,0).
  - ♦ If a line is centred, the feature value is one, else zero.
  - ♦ If a line contains at least one bold character, the feature value is one, else zero.

- **Linguistic information:**
  - ♦ If the word ‘abstract’ appears at the head of a line, the feature value is one, else zero.
  - ♦ If the word ‘keyword’ appears at the head of a line, the feature value is one, else zero.
  - ♦ In which class is each character to be categorized? Here, the classes are defined as ‘alphabet’, ‘number’, ‘hiragana character (Japanese text only)’, ‘katakana or kanji character (Japanese text only)’, ‘separator’ (e.g.)(‘,“”), ‘other marks’ (e.g. ~ { } &), ‘at mark’ (@), ‘period’, ‘blank’, and ‘comma’.

Evaluation Method

For classifying multi-classes using binary classifier SVM, we used the one-vs-rest classification method and a polynomial kernel of degree 2, which is defined by the following equation.

$$K(x_i, x_j) = (x_i \bullet x_j + 1)^d$$

We used YamCha[4] as an SVM learning package. We performed a five-fold cross validation test with the data in Table 1, in which English and Japanese papers were merged. Features and tags are given to SVM as follows: the four tags occurring before the target line; and the four features before the target line and the four features following it. These window sizes were determined by the results from a pilot study.

To investigate the effectiveness of the features, we conducted three tests with different combination of feature sets.

1. Information from text appearance
2. Linguistic information
3. A combination of 1 and 2

We use the following measures for evaluation.

$$\text{Accuracy 1=} \frac{\text{The number of correctly extracted fields}}{\text{The number of fields in the test sets}}$$

$$\text{Accuracy 2=} \frac{\text{The number of papers with correctly extracted fields}}{\text{The number of papers in the test sets}}$$

4.2.3. Results Results are shown in Table 3. As can be seen from the table, effective features are different for each field. For example, information from text appearance is more effective than linguistic information for the extraction of TITLE, TITLE_S, AUTHORS, and AUTHORS_S fields. On the other hand, linguistic information is more effective than information from text appearance for the extraction of EMAIL, KEYWORD, and KEYWORD_S fields. As a whole, the combination of text appearance and linguistic information improves the accuracy much more than using linguistic information or text appearance independently.

[4] http://cl.aist-nara.ac.jp/~taku/software/yamcha/

4.2.4. Discussion The reason that information from text appearance is effective for TITLE fields is that usually the largest fonts are used for the title of a paper, and the title is placed at the top of the first page with central alignment. However, linguistic information is also useful, because the accuracy of the combination of linguistic information and text appearance is higher than using only information from text appearance. This result also indicates that SVM is useful for selecting effective features.

Evaluation	Fields	The number of fields	Text appearance	Linguistic information	Appearance + linguistic
Accuracy 1	TITLE	945	0.959	0.884	0.972
	AUTHORS	940	0.817	0.767	0.899
	AFFILIATION	882	0.821	0.805	0.906
	EMAIL	323	0.538	0.960	0.960
	ABSTRACT	598	0.898	0.910	0.959
	KEYWORD	70	0.361	0.863	0.858
	OTHER	651	0.902	0.914	0.932
	TITLE_S	455	0.830	0.926	0.962
	AUTHORS_S	459	0.747	0.837	0.886
	AFFILIATION_S	426	0.809	0.834	0.892
	ABSTRACT_S	99	0.573	0.719	0.794
	KEYWORD_S	37	0.394	0.786	0.786
Accuracy 2		945	0.532	0.527	0.692

Table 3: Results of extracting bibliographic information from headers of papers

The reason that linguistic information is effective for extracting KEYWORD, KEYWORD_E, and EMAIL fields is that clue words such as ‘Keyword’ and ‘@’ are effective for the extraction. The accuracies for these fields using linguistic information alone are almost same as those for the combination of linguistic information and information from text appearance. It is assumed that SVM regards text appearance as ineffective for the extraction of KEYWORD, KEYWORD_E, and EMAIL.

4.3. Extraction of Bibliographic Information from Lists of References

To extract bibliographic information from lists of references, we make use of SVM. In this section, we first define the tags used in our examination. Second, we explain HMM-based models, which are the main approach in previous related studies. Third, we describe our SVM-based model. We compared our method with an HMM-based model by an experiment. Finally, we report the results.

4.3.1. Tag Definition We define a tag set for lists of references as follows;

- **AUTHORS** includes authors’ names. If a paper has multiple authors, AUTHORS tags are added before the first author and after the last author.
- **TITLE** includes the title of a paper. The TITLE tag excludes any quotation marks.
- **SOURCE** includes the source of a paper. The tag includes the title of conference proceedings, volume, number, publisher, URL and so on.
- **DATE** includes a publication year. DATE tags can also include a month or a day (e.g. ‘September 2003’).
- **PAGE** includes pages of a research paper (e.g. ‘pp. 1–8’, ‘p. 23’, ‘2138–2152’).
- **OTHER** includes other letter strings (e.g. ‘to appear’).
- **SEPARATOR** includes particular marks (e.g. comma and period).

A tagged example is shown as follows.

> [1] <AUTHORS>S. Lawrence, C.L. Giles, K. Bollacker</AUTHORS><SEPARATOR>, “</SEPARATOR><TITLE>Digital libraries and autonomous citation indexing</TITLE> <SEPARATOR>,” </SEPARATOR><SOURCE> IEEE Computer, vol.6, no.4</SOURCE> <SEPARATOR>,</SEPARATOR><PAGE>pp.67-71</PAGE><SEPARATOR>, </SEPARATOR><DATE>1999</DATE>.

4.3.2. Extraction based on Hidden Markov Model There have been several studies on extracting bibliographic information from lists of references based on Hidden Markov Models (HMM) (Borkar *et* *al.* 2001; Connan *et al.* 2000; Geng 2003; McCallum *et al.* 1999; Takasu 2003). We first briefly review the HMM-based extraction model.

A Hidden Markov Model uses the combination of two probabilities: the transition probability Pr($q_i \rightarrow q_j$) and the emission probability Pr($q_i \uparrow \sigma_k$). Both probabilities are defined by the following equations:

Probability of state transition

$$\Pr(q_i \rightarrow q_j) = \frac{c(q_i \rightarrow q_j)}{\sum_{q_i, q_j \in Q} c(q_i \rightarrow q_j)}$$

Probability of symbol emission

$$\Pr(q_i \uparrow \sigma_k) = \frac{c(q_i \uparrow \sigma_k)}{\sum_{\sigma_k \in \Sigma} c(q_i \uparrow \sigma_k)}$$

where $c(q_i \rightarrow q_j)$ is the number of transitions from state q_i to state q_j, and $c(q_i \uparrow \sigma_k)$ is the number of symbols emitted in state q_i. Both probabilities are calculated from a training data set. The sequence of states with the maximum score is calculated using the Viterbi algorithm.

The target language for related works is almost always English, and a unit of symbol emission and state transition is a word. However, our target languages are both English and Japanese, which has no boundaries between words. Even in English papers, blanks between words are sometimes undetected when PDF files are converted to XML files. We therefore apply characters as the unit of symbol emission and state transition.

- **Building an HMM Model**

We make use of a state transition model instead of an ergodic model that is a fully connected HMM, because state transition patterns are limited in our case. There are two ways for building a model: an automatic method and a manual method. McCallum proposed an automatic method that builds a model from observation of state transitions, where multiple neighbour states with the same class label were merged into one, and self-transition loop probability is increased (McCallum *et al.* 1999). However, their method tends to be too complex to create a valid model from a small amount of training data. We therefore build a state transition model manually. We first identify DATE and PAGE with simple regular expressions, and remove them from strings, because DATE and PAGE formats are almost fixed. Then we build the state transition model with other tags using the simple model shown in Figure 5. Only 1.4% (29/2077) of cases in our training data are not acceptable to this model. Table 4 shows accepted and rejected tag orders in lists of references.

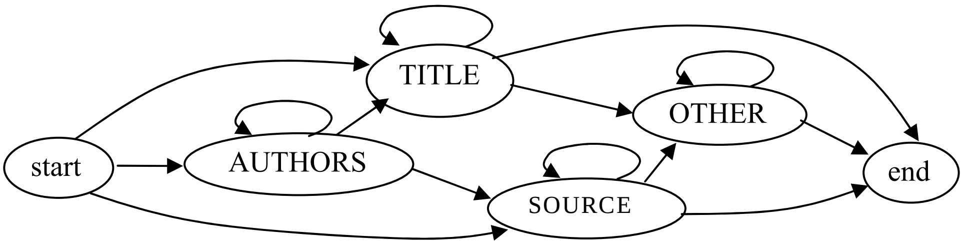


Figure 5: State transition model of HMM

Number of cases	Accepted order in our model	Number of cases	Rejected order in our model
1670	AUTHORS, TITLE, SOURCE	15	TITLE, AUTHORS, SOURCE
138	AUTHORS, TITLE, SOURCE, OTHER	6	AUTHORS, SOURCE, TITLE
		2	AUTHORS, TITLE, OTHER, SOURCE
107	AUTHORS, SOURCE		
40	AUTHORS, TITLE	1	TITLE, OTHER, SOURCE
38	TITLE, SOURCE	1	AUTHORS, SOURCE, TITLE, OTHER
23	SOURCE		
15	AUTHORS, SOURCE, OTHER	1	SOURCE, TITLE
6	AUTHORS, TITLE, OTHER	1	AUTHORS, OTHER, SOURCE
6	TITLE	1	AUTHORS, OTHER, TITLE, SOURCE
2	TITLE, SOURCE, OTHER		
2	SOURCE, OTHER	1	AUTHORS

Table 4: The occurrence order of field (except for DATE and PAGE)

4.3.3. Extraction based on SVM The SVM-based method assigns a class to each character in the same way as HMM-based method. Features and tags are given to SVM as follows: the k tags occurring before the target character; and the k features before the target character and the k features following it (Figure 6). We use a value of 13 for k in English texts, and six in Japanese texts. Values of k were determined in our pilot study.

To classify multi-classes based on a binary classifier SVM, we used the one-vs-rest classification method and a polynomial kernel of degree 3. We conducted an experiment by following two methods.

- SVM1: standard SVM
- SVM2: standard SVM, but DATE and PAGE fields are extracted first using regular expressions.

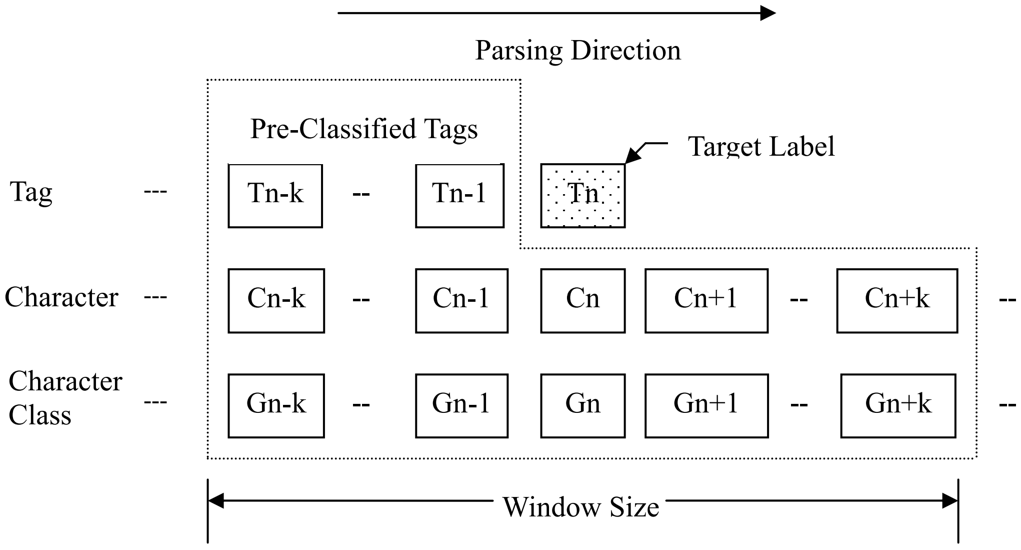


Figure 6: Features and Tags given to SVM

Evaluation Method

Sequences of characters with the same tags are combined into fields. Then, we evaluate by comparing the system output with the correct data using the following two measures.

$$\text{Accuracy 1} = \frac{\text{The number of correctly extracted fields}}{\text{The number of fields in the test set}}$$

$$\text{Accuracy 2} = \frac{\text{The number of bibliographic entries extracted correctly}}{\text{The number of bibliographic entries in the test set}}$$

We consider an entry correct if ‘alphabet’, ‘number’, and ‘hiragana, katakana and kanji characters (Japanese text only)’ in the system output match those in the correct data. An example of evaluation is shown in Figure 7. In this example, AUTHORS in the system output is correct, because the difference between the correct data and the system output is a separator ‘(’, while TITLE is incorrect, because the word ‘Models’ is identified as SOURCE in the system output, though it is TITLE in the correct data.

Correct	AUTHORS	S	DATE	S	TITLE		S
	J. Connan and C.W. Omlin	(	2000	)	Bibliography Extraction with Hidden Markov	Models	.
System output	AUTHORS Correct		DATE correct		TITLE Incorrect	SOURCE Incorrect	

Figure 7: Example of evaluation

We performed a five-fold cross validation test with the data in Table 1, in which English and Japanese papers were merged. Features, which are before and after four lines of a target line, and tags, which are before four lines of a target line, are given to SVM. These window sizes were determined in our pilot study.

4.3.4. Experimental Results The results are shown in Table 5.

Evaluation Measure	Field	English				Japanese			
		Number of elements	HMM	SVM1	SVM2	Number of elements	HMM	SVM1	SVM2
Accuracy 1	AUTHORS	1084	**0.909**	0.889	0.888	919	0.898	0.902	**0.908**
	TITLE	1044	0.795	0.812	**0.819**	883	0.826	0.836	**0.845**
	SOURCE	1100	0.740	0.765	**0.793**	923	0.772	0.788	**0.808**
	DATE	1061	**0.968**	0.887	**0.968**	853	**0.979**	0.945	**0.979**
	PAGE	652	**0.962**	0.933	**0.962**	465	**0.972**	0.933	**0.972**
	OTHER	106	**0.670**	0.538	0.642	64	**0.438**	0.406	0.359
	Total	5047	0.863	0.844	**0.874**	4107	0.872	0.867	**0.885**
Accuracy 2		1122	0.693	0.704	**0.733**	954	0.718	0.739	**0.764**

Table 5: Experimental results from extracting bibliographic information from lists of references

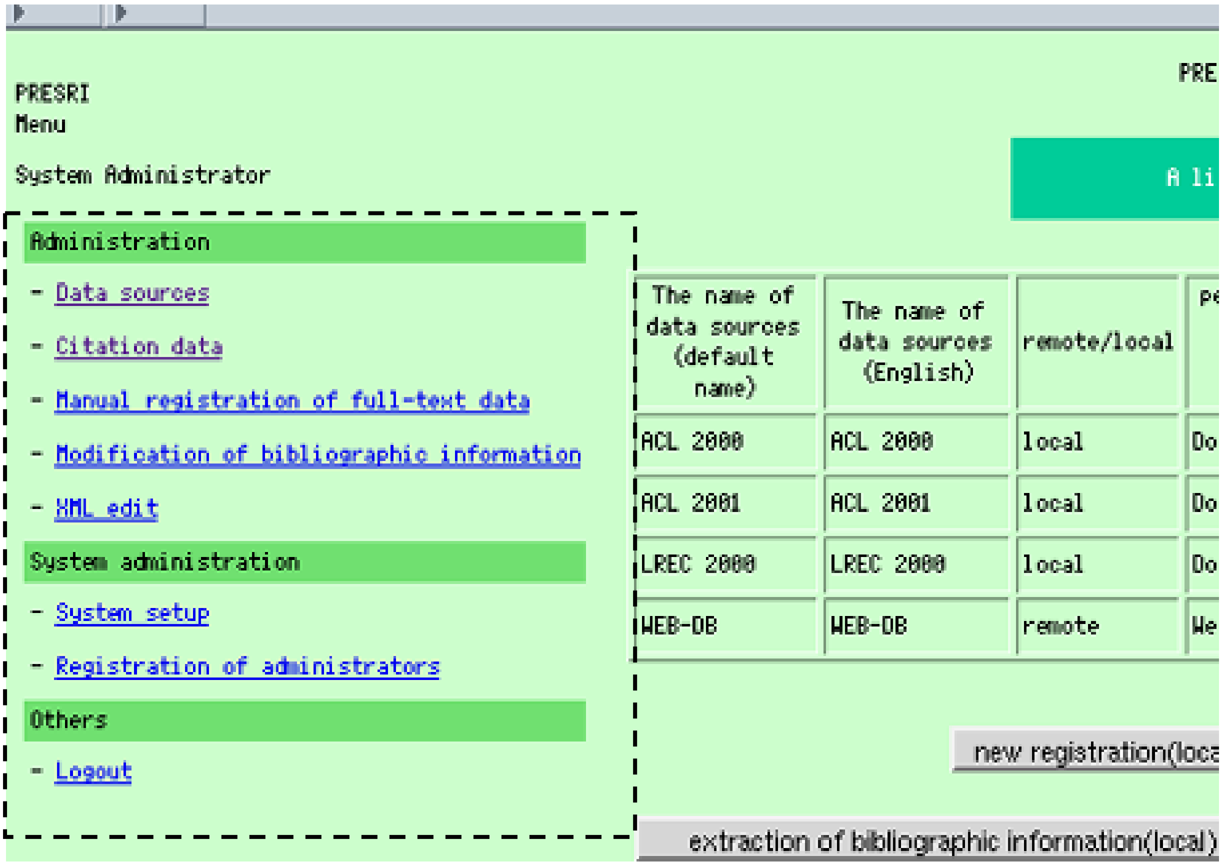
4.3.5. Discussion As can be seen from Table 5, performances of SVM2 in accuracy 1 and 2 are almost better than that of HMM. The main difference between the HMM-based method and SVM-based methods is that the latter methods take account of contextual information, i.e. all the features of characters within a window size and precedent tags. This information is considered to be effective for identification of fields. It is also notable that SVM2 is superior to other methods in both Japanese and English, with their very different linguistic information. From the results of the experiments, we can confirm that SVM2 is considered to be language independent and applicable to other languages.

5. System Behaviour

In this section, we introduce the behaviour of PRESRI with some snapshots.

5.1. Integration of Multiple Databases

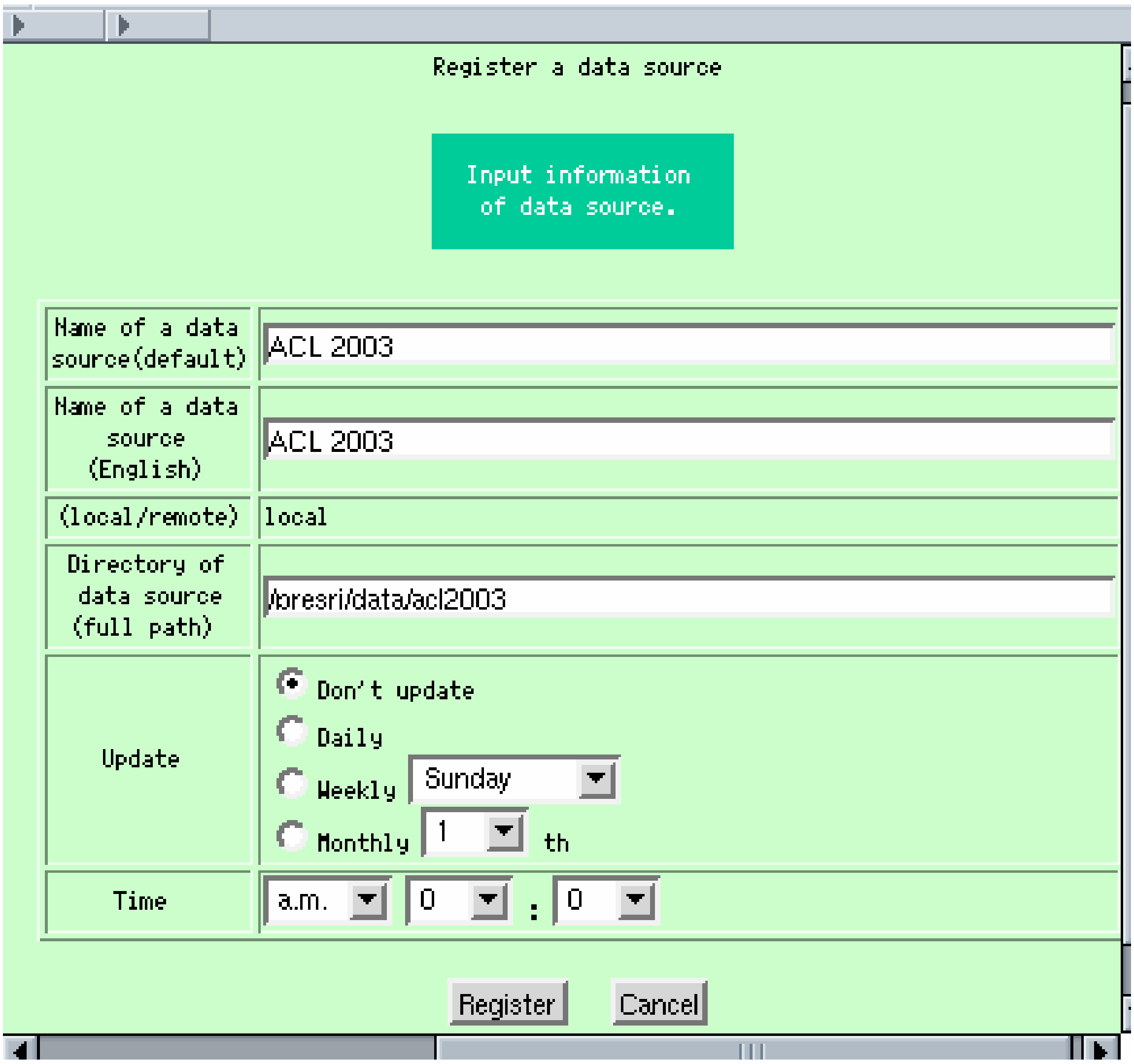
A PRESRI administrator can integrate local (e.g. CD-ROM) and remote (e.g. online DB) databases via a Web browser interface. First, the administrator enters a user ID and password to log into a PRESRI server. Then, a menu list (framed by a broken line in Figure 8), is shown. If the administrator selects an item in the list, the corresponding page is shown in the right frame. Figure 8 is the result after an administrator selected ‘Data sources’. Several databases were registered beforehand in this example. If the administrator selects ‘extraction of bibliographic information (local/remote)’, bibliographic information will be extracted from all the databases registered, and integrated on the local host. The administrator can register a new database on a local/remote host by selecting ‘new registration (local/remote)’. A new page for registration (shown in Figure 9) is displayed. In this page, the administrator fills in the name of the new database and URL, where the database exists[5]. The administrator also specifies an update interval.


PRE
PRESRI
Menu
System Administrator
A li
Administration
- Data sources
- Citation data
- Manual registration of full-text data
- Modification of bibliographic information
- XML edit
System administration
- System setup
- Registration of administrators
Others
- Logout
new registration(loca
extraction of bibliographic information(local)

The name of data sources (default name)	The name of data sources (English)	remote/local	pe
ACL 2000	ACL 2000	local	Do
ACL 2001	ACL 2001	local	Do
LREC 2000	LREC 2000	local	Do
WEB-DB	WEB-DB	remote	We

Figure 8: A list of registered databases

[5] To register a local database, the administrator enters the name of the directory containing the new database.



Register a data source

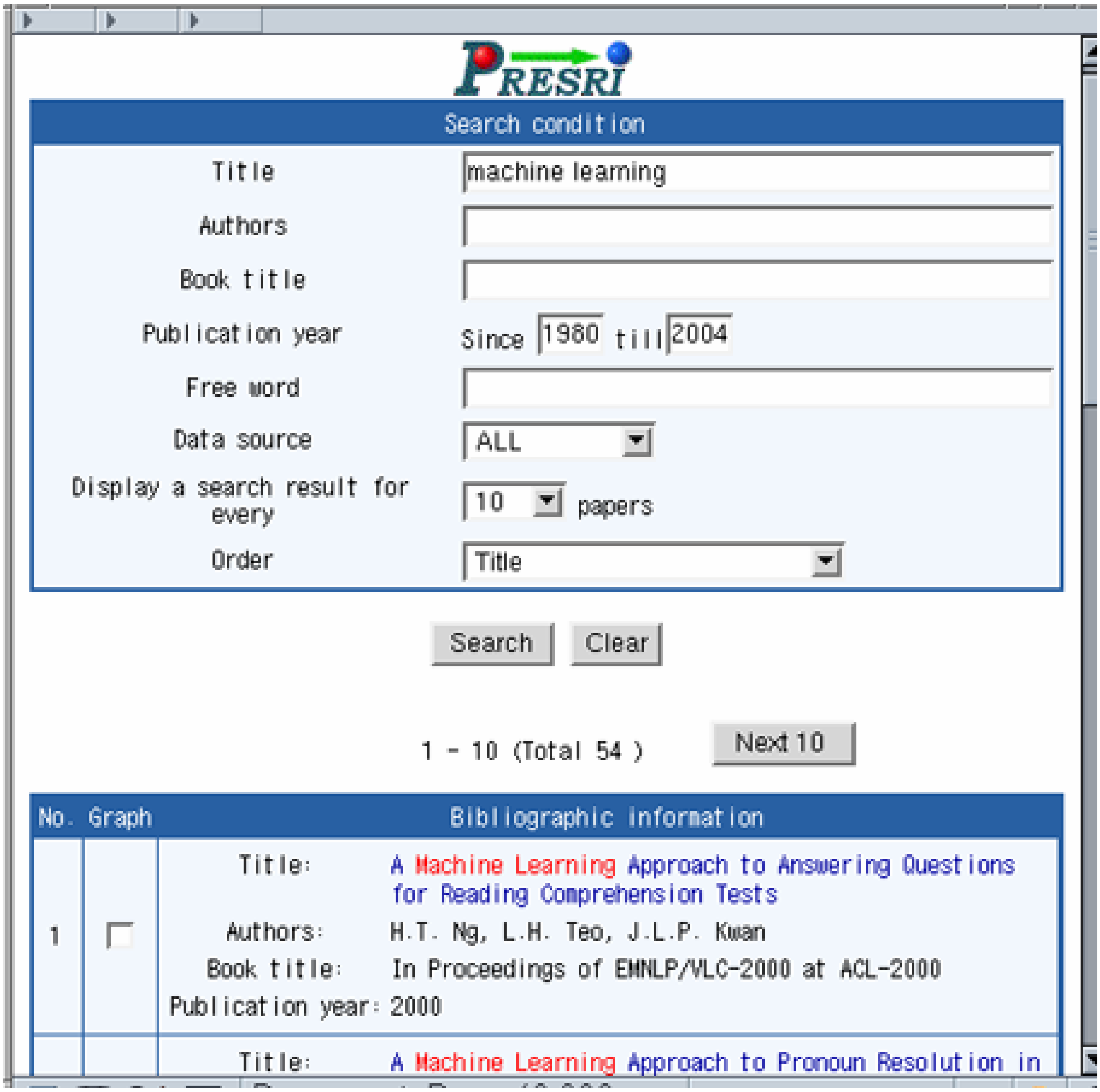
Input information of data source.

Name of a data source(default)	ACL 2003
Name of a data source (English)	ACL 2003
(local/remote)	local
Directory of data source (full path)	/oresri/data/acl2003
Update	● Don't update ○ Daily ○ Weekly Sunday ○ Monthly 1 th
Time	a.m. 0 : 0

Figure 9: Registration of a new database

5.2. Interface for Searching Papers

Figure 10 shows a search result for the key phrase ‘machine learning’. Check boxes are shown at the head of each result. If the user checks the boxes of relevant papers, and selects ‘display a citation graph’ at the bottom of the page, PRESRI shows the selected papers with some related papers visually, as shown in Figure 11.



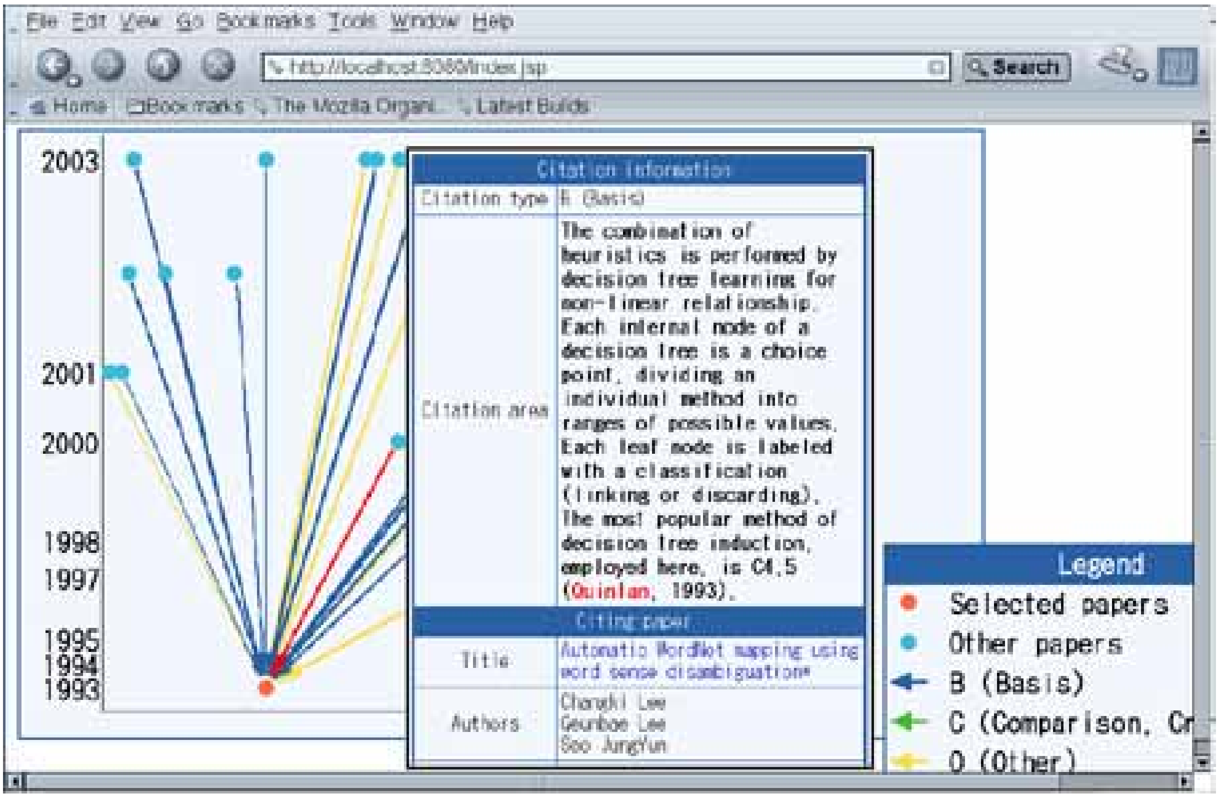
PRESRI

Search condition	
Title	machine learning
Authors	
Book title	
Publication year	Since 1980 till 2004
Free word	
Data source	ALL
Display a search result for every	10 papers
Order	Title

1 - 10 (Total 54)

No.	Graph	Bibliographic information	
1	□	Title:	A Machine Learning Approach to Answering Questions for Reading Comprehension Tests
		Authors:	H.T. Ng, L.H. Teo, J.L.P. Kwan
		Book title:	In Proceedings of EMNLP/VLC-2000 at ACL-2000
		Publication year:	2000
		Title:	A Machine Learning Approach to Pronoun Resolution in

Figure 10: Result of search by a key phrase ‘machine learning’



Citation information	
Citation type	B (Basis)
Citation area	The combination of heuristics is performed by decision tree learning for non-linear relationship. Each internal node of a decision tree is a choice point, dividing an individual method into ranges of possible values. Each leaf node is labeled with a classification (linking or discarding). The most popular method of decision tree induction, employed here, is C4.5 (Quinlan, 1993).
Citing paper	
Title	Automatic WordNet mapping using word sense disambiguation*
Authors	Changki Lee Geunbae Lee Seo JungYun

Figure 11: Display a graph of citation relationships and a citing area

In Figure 11, dots and arrows indicate papers and citation relationships between papers, respectively. Arrows are colour-coded according to their citation types. The title of a paper is shown in a pop-up window (Nanno *et al*. 2002), if the user puts the cursor over a dot (paper). If the user pauses the cursor for more than one second, authors and an abstract of the paper are shown together with the title. A citing area is shown in a pop-up window if a user puts the cursor over an arrow.

We can understand the progress and transition of ‘machine learning’ studies at a glance from the graph in Figure 11. We can also see the similarities and differences between papers by reading the citation areas. This information is helpful for an efficient survey of ‘machine learning’ studies.

6. Conclusions

In this paper, we introduced PRESRI, which can integrate multiple databases of research papers written in English and Japanese. For the purpose of constructing an exhaustive database, we first collected Postscript and PDF files on the Web, and constructed WEB-DB by extracting bibliographic information from the files. We then integrated several other databases with WEB-DB. We proposed a method for extracting bibliographic information from Postscript and PDF files based on the SVM. To investigate the effectiveness of our method, we conducted an examination, and found that our method is useful for both Japanese and English, and that our method is superior to HMM-based methods.

We also focused on the presentation of search results. We developed a presentation system for search results that makes it possible to understand the relationships between papers intuitively based on citation information. When a search operation is conducted, citation relationships with their citation types are shown visually as a search result.

Currently, our latest system is available at ‘http://www.presri.com,’ but the database is WEB-DB containing 20,000 papers only. In the near future, our system will be introduced to, and managed by, the library of Hiroshima City University and the Global Scientific Information and Computing Center in the Tokyo Institute of Technology. With a view to constructing a more comprehensive database, we will expand our system to deal with other languages.

7. Acknowledgements

This project was supported in part by IPA (The Information-technology Promotion Agency) under the grant for the 2002 Exploratory Software Project. The authors would like to express our gratitude to Dr. Masaru Kitsuregawa of the University of Tokyo for his suggestions for improving our system, and Mr. Syunichi Miyahara and Dr. Taizan Suzuki of DUO systems for their contributions to the development of our system.

References

Bergmark, D., Phempoonpanich, P. & Zhao, S. (2001). Scraping the ACM Digital Library. *SIGIR Forum*, 35(2), 1--7.

Borkar, V. Deshmukh, K. & Sarawagi, S. (2001). Automatic Segmentation of Text into Structured Records. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data* (pp. 175--186).

Connan, J. & Omlin, C.W. (2000). Bibliography Extraction with Hidden Markov Models. *Technical Report US-CS-TR-00-6.* Department of Computer Science, University of Stellenbosch.

Ding, Y., Chowdhury, G. & Foo. S. (1999). Template Mining for the Extraction of Citation from Digital Documents. In *Proceedings of the Second Asian Digital Library Conference* (pp. 47--62).

Herman, I., Melancon, G., & Marshall, M. (2000). Graph Visualization and navigation in information visualization: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 24--43.

Geng, J. (2003). Automatic Extraction and Integration of Bibliographic Information on the Web using Hidden Markov Models. Master's thesis. Department of Computer Science, Duke University.

Lawrence, S., Giles, L., & Bollacker, K. (1999). Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6), 67--71.

McCallum, A., Nigam, K., Rennie, J. & Seymore, K. (1999). Building Domain-specific Search Engines with Machine Learning Techniques. *Proceedings of the AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace*.

Nanba, H. & Okumura, M. (1999). Towards Multi-paper Summarization Using Reference Information. *Proceedings of the 16th International Joint Conferences on Artificial Intelligence,*(pp.926--931).

Nanno, T., Saito, S., & Okumura, M., (2002). Zero-Click: a System to Support Web Browsing. *The 11th International World Wide Web Conference*.

Seymore, K., McCallum, A. & Rosenfeld, R. (1999). Learning Hidden Markov Model Structure for Information Extraction. *AAAI 99 Workshop on Machine Learning for Information Extraction.*

Takasu, A. (2003). Bibliographic Attribute Extraction from Erroneous Reference based on a Statistical Model. *Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries 2003,* (pp.49--60).

Teufel, S. & Moens, M. (2000): What's yours and what's mine: Determining Intellectual Attribution in Scientific Text, *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.*

Teufel, S. (2001). Task-Based Evaluation of Summary Quality: Describing Relationships between Scientific Papers, *Proceedings of NAACL-2001 Workshop on Automatic Summarization.*

Vapnik, V.N. (1998). Statistical Learning Theory. New York: Wiley-InterScience.