Automatic annotation of bibliographical references in digital humanities books, articles and blogs

Young-Min Kim, Patrice Bellot, Elodie Faath, Marin Dacos

Automatic Annotation of Bibliographical References in Digital Humanities Books, Articles and Blogs

Young-Min Kim, Patrice Bellot
LIA, University of Avignon
339, chemin des Meinajaries
84911 Avignon, France
young-min.kim@univ-avignon.fr,
patrice.bellot@univ-avignon.fr

Elodie Faath, Marin Dacos
CLEO, Centre for Open Electronic Publishing
3, place Victor Hugo
13331 Marseille, France
elodie.faath@revues.org,
marin.dacos@revues.org

ABSTRACT

In this paper, we deal with the problem of extracting and processing useful information from bibliographic references in Digital Humanities (DH) data. A machine learning technique for sequential data analysis, Conditional Random Field is applied to a corpus extracted from OpenEdition site, a web platform for journals and book collections in the humanities and social sciences. We present our ongoing project with this purpose that includes the construction of a proper corpus and a efficient CRF model on this as a preliminary. This project is supported by Google Grant for Digital Humanities. A number of experiments are conducted to find one of the best settings for a CRF model on the corpus, and we verify them both in an automatic and manual way of evaluation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*Collection*

General Terms

Algorithms, Performance

Keywords

CRFs, Digital Humanities, Reference Annotation, Bibliography

1. INTRODUCTION

While primary research in digital humanities has mostly relied on the digitalization of the existing humanities texts, recent works are rather interested in combining technical tools into humanities data. Data visualization, user interaction and also information extraction in humanities data would be main examples in this recent trend. Among them, information extraction is a wide open field. And as in other disciplines applying computational methods such as in biology, we should first well define the problematics before applying techniques into data.

In this paper, we focus on the problem of extracting and processing useful information from bibliographic references in Digital Humanities (DH) data. The main interest of the bibliographic reference research is to provide automatic links between related references in citations of scholarly articles. The automatic link creation involves essentially the automatic recognition of reference fields, which consists of author, title and date etc. A reference now is considered as a sequence of these fields. Based on the correctly separated and annotated fields, different techniques can be applied for the creation of cross-links.

Most of the tools freely available process scientific references, by opposite to Digital Humanities, only and against a set of predefined patterns that are mostly regular expressions. For example, cb2bib [1] recognizes reference styles of the publications of the American Chemical Society and of Science Direct but do not work with the references we tried.

On the other side, some methods employ machine learning and numerical approaches, by opposite to symbolic ones that require a large set of rules that could be very hard to manage and that are not language independent. [1] cites the works of a) C.L. Giles et al. for the CiteSeer system on computer science literature that achieves a 80% accuracy for author detection and 40% accuracy for page numbers (1997-1999), b) Seymore et al. that employ Hidden Markov Models (HMM) that learn generative models over input sequence and labeled sequence pairs to extract fields for the headers of computer science papers, c) Peng et al. that use Conditional Random Fields (CRFs) [3] for labeling and extracting fields from research paper headers and citations. Other approaches employ discriminatively-trained classifiers such as SVM classifiers. Compared to HMM and SVM, CRF obtained better labeling performance [4] (about 99% for author identification and for date, 94% for titles, 87% for Editor...). Some recent papers propose methods to disambiguate author citations [2, 7] or geographical identifiers [8]. These state of the art approaches seem to achieve good results but they proceed on limited size collections of scientific research papers only and they do not resolve all the difficulties we identified above.

Here we choose CRFs as method to tackle our problem on the bibliographic references extraction on DH data. It is a

[1] http://www.molspaces.com/cb2bib/

BooksOnline'11, October 24, 2011, Glasgow, Scotland, UK.

type of machine learning technique applied to the labeling of sequential data. The discriminative aspect of this model enables to overcome the restriction of previously developed HMM [5], then provides successful results on reference field extraction [4]. However, most of the earlier studies deal with relatively well structured citation data with simple format such as scientific articles. Besides, DH reference data generally includes a lot of less structured bibliographical parts and various different formats.

We started up a research project, *Robust and Language Independent Machine Learning Approaches for Automatic Annotation of Bibliographical References in DH Books, Articles and Blogs*, supported by Google Grant for Digital Humanities on March 2011 to approach this problem. It is a R&D program for in-text bibliographical references published on CLEO's OpenEdition platform. The program allies Cléo to the Laboratoire Informatique d'Avignon (LIA), and aims to construct a software environment enabling the recognition and automatic structuring of references in academic digital documentation whatever their bibliographic styles. Over time the tool will enable the development of cross-linking functions within the platform to outside sources.

We first give an overview of our system for the automatic annotation of bibliographical references in DH documents and also a brief explanation of the main tool, CRFs in Section 2. We then describe our corpus extracted from the Revues.org site that consists of manually identified and annotated references (Section 3). In Section 4, we detail the process of the construction of an efficient CRF model adapted to our corpus. We empirically evaluate the CRF model via a number of experiments then give a conclusion and future work (Sections 5 and 6).

2. BIBLIOGRAPHICAL REFERENCE ANNOTATION

One of the final goals of the project is to construct an automatic structuring of references in the DH site OpenEdition [2], composed of three different platforms, Revues.org, Hypotheses.org and Calenda. These platforms are dedicated to electronic resources in the humanities and social sciences. As a primary work, we automatically label the reference fields in the articles of Revues.org site. This automatic detection will be integrated into a system which makes automatic cross-linking in citations.

Meanwhile, the use of the detected reference fields is not only restricted to the creation of automatic cross-linking. The annotated bibliographical information can be used for the information retrieval in the articles of both Revues.org and Hypotheses.org sites among which the latter is a platform for scholarly blogs open to the academic community in all disciplines of the arts, humanities and social sciences. We also expect that this extracted information can be moreover used for the IR in other external platforms.

Our automatic system for the bibliographic reference extraction is called Bilbo, the combination of two essential terms for the system, bibliography and robot. Bilbo will be freely available as soon as we establish a dependable system.

2.1 Conditional Random Fields

Automatic annotation can be realized by building a CRF model. A CRF is a discriminative probabilistic model developed for labeling of sequential data. Compared to a hidden Markov model (HMM), one of the traditional approaches for sequential data, it is better adapted to rich characteristics of input data. This advantage essentially allows a CRF to well model the sequences of bibliographical fields by including a lot of features, which represent the characteristics of input.

General CRFs are targeted not only at sequential data but also at general graph data, whereas a special version, linear-chain CRFs are mainly applied to sequence labeling. Therefore, we apply especially a linear-chain CRF to our reference labeling problem as in the literature. As we mentioned above, the core advantage of a CRF model in contrast with a HMM model comes from its discriminative aspect in modeling. By definition, a discriminative model maximizes the conditional distribution of output given input. So, if we construct a CRF model based on this conditional distribution, any factors dependent only on input are not considered as modeling factors, instead they are treated as constant factors to output [6]. This aspect derives a key characteristic of CRFs, the ability to include a lot of input features in modeling. It is essential for some specific sequence labeling problems such as ours, where input data has in general rich characteristics. The conditional distribution of a linear-chain CRF for a set of label $\mathbf{y}$ given an input $\mathbf{x}$ is written as follows :

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\}, \qquad (1)$$

where $\mathbf{y} = y_1 ... y_T$ is a state sequence, interpreted as a label sequence, $\mathbf{x} = x_1 ... x_T$ is an input sequence, $\theta = \{\theta_k\} \in R^K$ is a parameter vector, $\{f_k(y_t, y_{t-1}, \mathbf{x}_t)\}_{k=1}^{K}$ is a set of real-valued feature functions, and $Z(\mathbf{x})$ is a normalization function. Instead of the word identity x_t, a vector $\mathbf{x}_t$ is substituted in feature functions, because the vector contains all information about the word.

A feature function often has a binary value, which is a sign of the existence of a specific feature. A function can measure a special character of input token x_t such as capitalized word. And it also measures the characteristics related with a state transition $y_{t-1} \rightarrow y_t$. Thus in a CRF model, all possible state transitions and input features including identity of word itself are encoded in feature functions.

Inference of CRFs is conducted by introducing the Viterbi algorithm for computing the most probable labeling sequence, $\mathbf{y}^* = \arg\max_y P_\theta p(\mathbf{y}|\mathbf{x})$ and the forward-backward algorithm for marginal distributions. It is used for the labeling of new input observations after constructing a CRF model, and also applied in learning process to compute parameter values. Parameters are estimated by maximizing conditional log likelihood, $l(\theta) = \sum_{i=1}^{N} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$ for a given learning set of N samples, $D = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$.

3. CORPUS PREPARATION

Faced with the great variety of bibliographical styles present on the three platforms developed by Cléo and the dissemination of references within texts, we have implemented a series of stages corresponding to the various issues encountered on the platforms. In this section, we detail the nature of Revues.org data that justifies our methodology to the creation of corpus. In brief, we construct three different types of corpus manually annotated using TEI guidelines.

[2]http://www.openedition.org/

3.1 Revues.org document characteristics

Revues.org is the oldest French platform of academic journals online. It now offers more than 250 journals available in all of the disciplines of the humanities and social sciences, with predominance of history, anthropology and sociology, geography and archaeology. Beyond the commitment in favor of open access (more than 40,000 articles in open access), the platform is based on a model of appropriation of the electronic publishing process by publishers and producers of content. The online publication is made through the conversion of articles into XML TEI format and then into XHTML format and allows the viewing of the full text in web browsers. The specific technical quality needed for the publishing of scientific texts is provided by many functions: metadata management, multiple indexes, management of endnotes, automatic table of contents, numbering of paragraphs and attribution of DOI.

We expect that our work on bibliographical references will enrich the Revues.org platform via a number of possible services derived from basic auto-labeling operation as explained in Section 2. However, there are many difficulties in the treatment of DH articles. One main reason is the diversity of source disciplines that makes various styles in reference formatting. Moreover, even on a same discipline or journal, we can easily find quiet different reference forms. Another important difficulty compared to scientific research papers is that the references of DH are not always at the end of article, in a bibliography part. They sometimes arise in body of the article or in footnote.

Against these difficulties, we first well define TEI XML tags for the bibliographical parts. Then the references are manually annotated using the defined XML tags. As a solution to the diversity of the reference types, we constructed three different corpus according to the types.

3.2 Manual annotation of Revues.org data

TEI and annoatation of bibliographic references

TEI [3] is a markup language for describing features of texts, and in our case, fields of bibliographic references. Three levels of description are possible with this language:

- <bibl> : for all bibliographic elements
- <biblStruct> : structure the reference with predefined elements and we find this model on HAL or TEL
- <biblFull> : this model uses only elements allowed under <fileDesc>

In our corpus, we use the standard description <bibl> to annotate freely the bibliographic references. Indeed, OpenEdition presents a variety of bibliographic styles that <biblStruct> or <biblFull> can not describe. Other reason is that this standard description can be adapted for special references, for example, the case of inclusion or to indicate that reference is a working paper or published in a scientific event. But, some references present different peculiarities and require a set of internal links.

Three different levels of corpus

From the digital resources of Cléo, we identified and selected several references with different bibliographic styles. We correctly described these references in order to build a corpus

[3]http://www.tei-c.org/

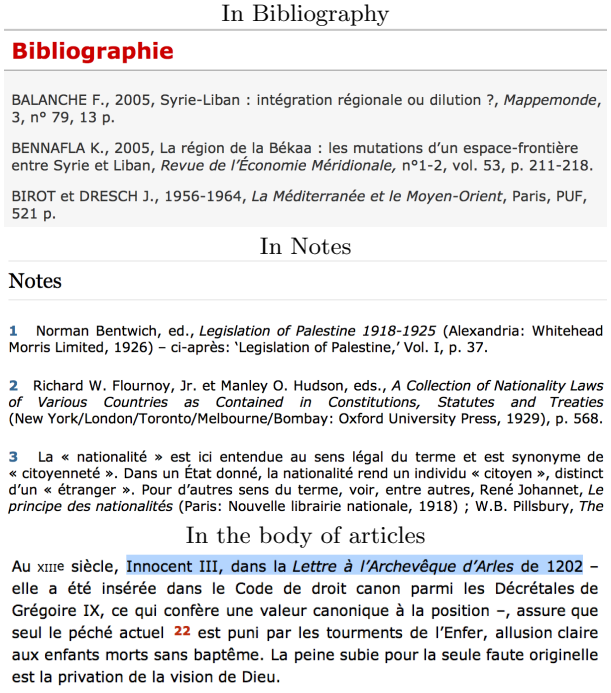

Figure 1: Different styles of bibliographic references

for Bilbo. OpenEdition represents more than 70,000 documents allocated to three platforms. All these documents have specific bibliographic styles imposed by the scientific journals or adopted by the author of the article. We can distinguish three levels of difficulties for identification and annotation of references :

- Level 1: the references are at the end of the article in a heading "Bibliography". Manual identification and annotation are simple.
- Level 2: the references are in footnotes and they are less formulaic compared to level 1 but standardized.
- Level 3: the references are in the body of articles. The identification and annotation are complex. Even finding the begins and ends of bibliographic references is difficult.

Figure 1 picks the examples of these levels that constitute three different corpus. As we can see in the examples, not all the notes include bibliographic information.

Thus the constitution of the corpus level 1 is based on the Revues.org site articles having the bibliography part at the end. Ultimately, 32 journals were selected and 38 sample articles have been taken considering the diversity of styles : 737 bibliographic references have been identified and annotated using TEI language. Table 1 shows an example of manually annotated reference in the corpus level 1, which our primary experiments principally target at.

In the second level of corpus, the references are in the footnotes and standardized. We annotate references using same tags to the first corpus. An important character of the corpus level 2 is that it contains link information between references on notes. That is, several references are shorten including just essential parts such author name, but

Table 1: An example of reference in corpus level 1

<bibl><author><surname>Arcelin</surname>, <forename full= "init">P.</forename></author><c type= "comma">,</c><author><surname>Congès</surname>, <forename full= "init">G.</forename></author> et <author><surname>Willaume</surname>, <forename full= "init">M.</forename></author><c type= "comma">,</c> <edition><date>1990</date></edition><c type= "point">.</c> <title level= "m">Compte-rendu des recherches archéologiques à Entremont (1988-1989)</title><c type= "comma">,</c> <distributor>ministère de la Culture et de la communication (sous-direction de l'Archéologie)</distributor><c type= "point">.</c></bibl>

sometimes are linked to other references, which have more detailed information on the shorten references. This case often occurs when a bibliographic document is referred more than a time. The links are established through several specific terms as supra, infra, ibid, op. cit., etc. In this case, we assigned a unique identifier to highlight the links in the documents. We selected 29 journals from a stratified selection and we extracted 30 articles after analysis of document.

Actually, we are finishing the primary experiments on corpus 1 and starting modeling on corpus 2 and the manual annotation of corpus level 3 is just started. This paper reports especially our corpus preparation and the experiments on the first level of corpus. In the following section, we explain the detail of the modeling process on the corpus level 1. By repeating many experiments, we could establish an efficient way to set the features and labels for learning data.

4. MODEL CONSTRUCTION

To apply a probabilistic model on a corpus, it is indispensable to well determine a learning data format. As our corpus contains relatively complex information compared to other data in the literature of reference field extraction and output labels are not determined, we need a learning data extraction process before applying a CRF model on the corpus. Our first work especially concentrates on the preparation of an appropriate learning data. Then we learn a CRF model on this newly prepared data and verify its performance.

In this section, we explain the process of formatting learning data. It includes a brief analysis of the characteristics of Revues.org reference corpus manually annotated. Then we build a CRF model on the prepared learning data using an existing language processing toolkit, MALLET software [4], developed by Andrew McCallum and his team members of Umass Amherst university.

4.1 Corpus character analysis for learning set

The necessary data to learn a CRF model are input sequences, output labels and input features. A reference string is segmented into a sequence of tokens, where a token is identified by a whitespace character or a tag. One of the difficulties in the preparation of learning data from corpus level 1, xml source data is to determine output labels and input features. Since not all the tags are appropriate as labels, and some attributes are good for labels, we have to make a decision for the selection of output label types. In the same way, we should select the suitable input features.

In Table 2, we compare our corpus level 1 with a standard reference dataset, Cora [4]. This comparison allows a better understanding of the complexity of our corpus and it also gives us some idea of which labels and features should be selected. Main differences are the number of tags in Revues.org corpus much more than that of Cora, and the detailedness of author annotation involving the separation between individual authors and even name types. The complexity of data structure in our corpus results from the design basis of the DH reference corpus. We tried to rather fully annotate the references considering reuse of corpus, than make annotation fits perfectly for the reference field extraction task.

[4]http://mallet.cs.umass.edu/

Table 2: Cora data and Revues.org corpus level 1

Cora reference data	Revues.org corpus level 1
500 references	737 references
13 fields(tags) of reference	30 tags of reference
Fields = labels	Labels are not decided
No multi labels	Multi levels tag structure
No separation of authors	Authors are separated, surname and forename also
Features are not verified	Features are not verified
22 features in the article	Need to determine features
No separation on punctuation	Important punctuation marks are annotated

Table 3: Cora data label and Revues.org corpus tag

Cora	Revues.org
<author>	<author>, <surname>, <forename>
<booktitle>	<title> of <relatedItem>+ attributes in title or in bibl
<date>	<date>
<editor>	<editor>, <author> of <relatedItem>
<institution>	<orgName>
<journal>	<title> of <relatedItem>+ attributes in title or in bibl
<location>	<pubPlace>, <country>
<note>	
<pages>	pp attribute of <bibleScope>
<publisher>	<publisher>
<tech>	thesis or technical report etc. attributes in title or in bibl
<title>	<title>
<volume>	vol attribute of <bibleScope>

4.2 Labels and features

Table 3 compares Revues corpus tags to Cora data labels. According to the TEI, the <relatedItem> tag indicates a book, a collection, or a journal etc. where the referred article is published. Several attributes seem better for labels besides not all the tags are appropriate as labels.

A way to choose label types is using all the tags as labels. In this case, we have two major problems to do that : a token sometimes has multi-tags and there are many meaningless tags for labels. Therefore, we first try a simple method for choosing labels composed by the following rules.

- Choose the closest tag for a token
- Choose the upper tag if the closest tag of the token is one of <hi>, <abbr>, <pb>, <ptr>, <lb> or <emph>

Then we conduct many experiments by modifying the label selection criteria and finally got a set of optimized rules and the determined labels in current version are described in Table 4.

The feature manipulation is also very important for constructing an efficient CRF model. As in the label selection,

Table 4: Labels for learning data

Labels	Description
surname	surname
forename	forename
title	title of the referred article
booktitle	book or journal etc. where the article is published
date	date, mostly years
publisher	publisher, distributor
c	punctuation
place	place : city or country etc.
biblscope	information about pages, volumn, number etc.
abbr	abbreviation
orgname	organization name
nolabel	tokens having no label (to be modified maybe)
bookindicator	the word "in" when a related reference is followed
extent	total number of page
edition	information about edition
name	editor name : confused with surname and forename
pages	pages, in this version we don't use it
OTHERS	rare labels such as genname, ref, namelink, author, region

we also explored the effects of features concerning the characteristics of token via a number of experiments. Sometimes too detailed features rather decrease the performance of CRF, so we need a prudent selection process of features. Our currently selected features and their descriptions are presented in Table 5.

Table 5: Features for learning data

Feature name	Description
ALLCAPS	All characters are capital letters
FIRSTCAP	First character is capital letter
ALLSAMLL	All characters are lower cased
NONIMPCAP	Capital letters are mixed
ALLNUMBERS	All characters are numbers
NUMBERS	One or more characters are numbers
DASH	One or more dashes are included in numbers
INITIAL	Initialized expression
WEBLINK	Regular expression for web pages
ITALIC	Italic characters
POSSEDITOR	Possible for the abbreviation of editor

4.3 Tokenization

Recall that our manual annotation with TEI guidelines includes the annotation of some important punctuation marks (Table 2), which are used as tokens. But as we do not have this kind of information for a new reference to be estimated in real world, learning and testing with these tokens do not reflect an accurate performance of a CRF model. Therefore, we need to newly tokenize the corpus as if we do not have any additional information than whitespaces. The punctuation marks can be treated as either individual tokens or as the attached features to the previous or next word.

In both cases, we need to make a supplementary decision on labels and features. For example, when we decide to separate all the punctuation marks as tokens, we should also decide which labels the marks will have. We can simply distribute an identical label such as "punctuation" to them or maybe make some categories of punctuation to label them. In order to acquire the most reasonable learning data in terms of tokenization, features and labels. We constructed about 40 different CRF models changing the criteria of tokenization, especially punctuation treatment, and the types of features and labels. Our final punctuation criteria is to tokenize all the punctuation marks and label them with a specific label.

Figure 2 describes the flattened structure of a reference in our corpus level 1, and the version of learning data where 22 label types and 11 features are used. On the left side, each line represents a token where the word identity is on the very left, its tag is on the right end, and the tag attributes are surrounded by double plus signs. Learning data is likewise represented in the order of token, features (capitalized characters), and label.

```
BOSERUP ++ ++    author surname
!NONE!   ++ ++   author
E. ++  init ++    author forename
, ++  comma  ++    c
1965 ++ ++    edition date
, ++  comma  ++    c
The ++  italic  m ++    hi title
conditions ++  italic  m ++    hi title
of ++  italic  m ++    hi title
Agricultural ++  italic  m ++    hi title
Growth ++  italic  m ++    hi title
: ++  italic  m  ++    hi title c
The ++  italic  m ++    hi title
Economics ++  italic  m ++    hi title
of ++  italic  m ++    hi title
Agrarian ++  italic  m ++    hi title
Change ++  italic  m ++    hi title
under ++  italic  m ++    hi title
Population ++  italic  m ++    hi title
Pressure ++  italic  m ++    hi title
, ++  comma  ++    c
Aldine ++ ++    publisher
, ++  comma  ++    c
Chicago ++ ++    pubplace
, ++  comma  ++    c
218 ++ ++    extent
p ++ ++    abbr
. ++ ++    abbr c
```

```
BOSERUP ALLCAP surname
E. ALLCAP INITIAL forename
, c
1965 ALLNUMBERS date
, c
The ITALIC FIRSTCAP title
conditions ITALIC ALLSMALL title
of ITALIC ALLSMALL title
Agricultural ITALIC FIRSTCAP title
Growth ITALIC FIRSTCAP title
: ITALIC c
The ITALIC FIRSTCAP title
Economics ITALIC FIRSTCAP title
of ITALIC ALLSMALL title
Agrarian ITALIC FIRSTCAP title
Change ITALIC FIRSTCAP title
under ITALIC ALLSMALL title
Population ITALIC FIRSTCAP title
Pressure ITALIC FIRSTCAP title
, c
Aldine FIRSTCAP publisher
, c
Chicago FIRSTCAP place
, c
218 ALLNUMBERS extent
p ALLSMALL abbr
. c
```

Figure 2: Flattened structure of corpus(left) and the current version of extracted learning data for Revues.org corpus level 1(right)

5. EXPERIMENTS

In this section, we describe our experimental result on corpus level 1. As explained above, we constructed more than 40 CRF models applying different learning datasets with various tokenization, labels and features. The very first two or three experiments aimed at two main objectives, verifying the suitability of CRFs on our task and signposting the various directions for the preparation of an appropriate learning dataset. With this objectives, we first started with a simple learning dataset where the input sequences are automatically extracted from the corpus with its internal tokenization manually done. After verifying that this trial gives a reasonable result in terms of labeling accuracy, we gradually added the extraction rules for labels and features.

We do not record all the remarkable discoveries during the experiments in this section, but present the result with some important experimental settings including the final one at this stage. Four different CRF models are selected to review our works. After eliminating some erroneous references, we've got 715 references prepared for learning and testing. 70% of them (500 reference) are used as learning data, and the remaining 30% (215 references) are used as test data. The identical tokenization technique, labels and features are applied to both datasets on each experiment.

5.1 Measures

We evaluated the auto-labeling result based on the ground truth method, which means that we compare the estimated labels of test references with the true labels of them. For an accurate evaluation, both automatic and manual evaluation methods are applied. We used the micro averaged precision and recall as the former and also manually evaluated in detail considering the correctness level of each estimated reference. The latter approach contributes also to the evaluation of the learning data.

Micro averaged precision and recall

As the measures, we used the micro-averaged precision, which computes the global accuracy of the estimated result, and also the precision and the recall of each type of labels. For the micro-averaged precision, we count all the correctly estimated tokens regardless of the type of labels and divide it by the total number of estimated tokens. The precision of a type of label is the proportion of the correctly estimated tokens in all the tokens estimated as the label. The recall of a type of label is the proportion of the correctly estimated tokens in all the tokens having originally the label.

Manual evaluation

Since some labels are automatically extracted from corpus and also there are some miss annotation in the corpus, our learning data can not be perfect. We expect that the manual evaluation correct this errors. The more important aim of the manual evaluation is to find the erroneous patterns made by Bilbo system. The found patterns will contribute the system rebuilding by modifying the learning data or the CRF model itself.

5.2 Evaluation and data verification

The overall auto-labeling accuracies are represented in Table 6. We compared five different CRF models learned with different data settings. The criteria for the extraction of learning data is described in the table.

The result on the first stage confirms that a learned CRF with our first trial version without any preprocessing gives a reasonable estimation accuracy (85.34% in general accuracy) on a test set. It is encouraging for applying a CRF model to our task, because we did not use any local, layout or external features. When we look inside of the learning data of the first version, several meaningless tags chosen as labels, such as <lb>, <pb>, <ptr> and <emph>, occur very often inside of the other meaningful tags. These tags are verified to give low performance in terms of both precision and recall, so we replace them with its upper tags. The 15th stage gives the most effective learning data among our experiments when not considering the tokenization problem. In this setting, some useless tags as labels are eliminated, and also two simple features, 'comma' and 'point' are introduced to describe the nature of punctuation. With this learning data, we've got 88.54% in general accuracy.

On the remaining stages, we applied various tokenization technique. As a result, separation all the punctuation marks as tokens works well, especially when the marks are all labels with a same one. In the $21th$ stage, the overall accuracy increased again as 89.56% with this punctuation treatment. But we found a problem of <title> tag, because current labeling system can not distinguish the main title of article and the book title where the article is published. This is caused by our manual annotation where we wrote down the character of title in the attributes of tags.

In the $28th$ and $35th$ stages, we extracted the nature of title information from the corpus. This is not always easy because there are several attributes indicating both article title and book title. Considering the attributes and the place of title etc., we successfully separated <title> and <booktitle> for learning and test data. Of course the accuracy is decreased compared to the $21th$ stage because of the more detailed labels in the 28th stage. However, by introducing appropriate features, we finally get 88.23% of overall accuracy on the test dataset.

Figure 3 shows the detailed performance of $35th$ stage model, our current final version on the corpus level 1. This result is quiet encouraging because the most important three labels, surname, forename and title give about 90% of precisions and recalls.

Accuracy (Micro Averaged Precision)

5868/6651*100 = **88.23 %**

********* Precision *****				********* Recall *****			
surname	**314**	**336**	**93.45**	**surname**	**314**	**352**	**89.20**
forename	**274**	**303**	**90.43**	**forename**	**274**	**320**	**85.63**
title	**1945**	**2312**	**84.13**	**title**	**1945**	**2085**	**93.29**
booktitle	201	293	68.60	booktitle	201	418	48.09
publisher	282	342	82.46	publisher	282	373	75.60
date	239	286	83.57	date	239	258	92.64
place	155	212	73.11	place	155	168	92.26
biblscope	104	125	83.20	biblscope	104	140	74.29
abbr	124	146	84.93	abbr	124	138	89.86
nolabel	62	98	63.27	nolabel	62	104	59.62
edition	9	13	69.23	edition	9	71	12.68
orgname	18	19	94.74	orgname	18	42	42.86
extent	30	30	100.00	extent	30	31	96.77
name	5	18	27.78	name	5	27	18.52
bookindicator	21	22	95.45	bookindicator	21	22	95.45
namelink	5	5	100.00	namelink	5	5	100.00
				ref	0	5	0.00
				author	0	2	0.00
				genname	0	1	0.00
				region	0	1	0.00
c	2080	2091	99.47	c	2080	2088	99.62

Figure 3: Detailed performance of current version of CRF model on corpus level 1

This original corpus, which is manually annotated, contains rich information about references. But as our reference field identification needs comparably simple labeling structure considering the limitation of an automatic learning system, we extracted an appropriate learning data from the original corpus. Automatic extraction of learning and test data from the corpus could not be perfect because of the complexity of original corpus and the possible errors of manual annotation. During a number of experiments conducted to construct an effective learning data, we verified that several estimation errors come from some miss-annotated tokens in learning data.

To prevent this kind of problems, we examine manually the completeness of learning and test data in terms of correct labeling. This examination is in fact organized a part of our manual evaluation. So in our manual evaluation, we have two objectives : evaluation in detail the auto-labeling result by current version of Bilbo from the experiment stage 35, and verification of the completeness on the learning and test dataset. This verification accompanies the correction of erroneous manual annotation in dataset.

In order to detect the annotation error of Bilbo, we cat-

Table 6: Overall accuracies of the constructed CRF models with different learning data

Stage	Tokenizing	Labels	Features	Accuracy	Remarks
1	Based on manual annotation and whitespaces	The most nearest tag for each token (28 tags)	No features	85.24%	Not applicable for new reference (because of manual tokenization).
15	same as above	Elimination of some rare and inappropriate tags etc.	comma, point	88.54%	same as above
21	Tokenize all the punctuation marks	Punctuation are labeled as <c>	No features	89.56%	No separation between title and booktitle.
28	Tokenize all the punctuation marks. Initial words are first found, treated as a token	Separation of <title> and <booktitle>	6 features	86.32%	Separation of title and booktitle. Total number of tokens decreases using initial.
35	**same as above**	**Separation of <title> and <booktitle>; Unifications of similar tags**	**11 features**	**88.23%**	**Separation of title and booktitle. Total number of tokens decreases using initial.**

egorize the estimated reference into three groups: perfectly labeled, partially labeled and wrongly labeled. The evaluator, who initially annotated the corpus and is specialist in the Humanities domain, strictly qualified the result estimated by Bilbo. If there are one or more mistakes but not ruin much the labeling result, we mark the reference as partially labeled. But if there are many mistake which effects strongly the quality of labeling, we mark it as wrongly labeled. The categorized result is presented in the left side of Figure 4.

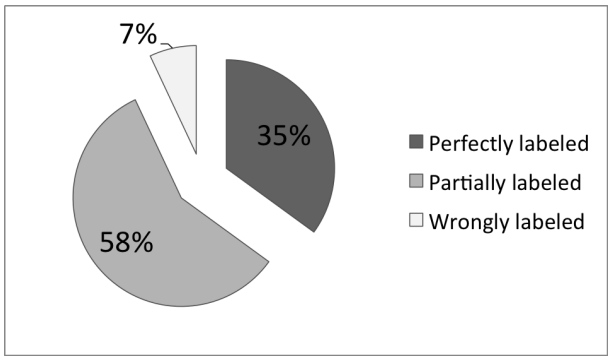

Figure 4: Manually evaluated labeling quality

We also verified from which labels the errors come. As in Figure 5, there are about five big categories: author, title, publication information such as pages, punctuation and others. More detailed analysis on the types of errors and the mistake patterns of Bilbo are all recorded by the evaluator, and this will be used for the modification of Bilbo system.

6. CONCLUSION AND PERSPECTIVE

We have presented the automatic reference labeling system, Bilbo on CLEO's OpenEdition platform. We constructed an efficient CRF model on the corpus level 1, which deals with the well organized bibliography part at the end of articles in Revues.org. Because of the difficulties in DH articles, we concentrated in analyzing in detail the characteristics of the references. The experimental results confirm that the utilization of CRFs to our task is effective and also our efforts to find a well-defined learning dataset were useful. A remarkable difference of our work compared to the state of the art, is that we could successfully separate the individual authors and even their surname and forename. It

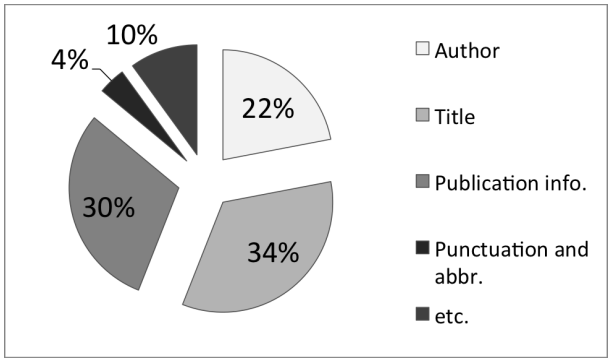

Figure 5: Current Bilbo system error sources

is important because the final objective of the system is not restricted to the extraction of reference fields but it includes also the development of possible services using this extracted information A more delicate separation in the author fields will allow the more useful services.

The work presented here is the beginning of our entire information extraction system on the OpenEdition platform. Our future works can be divided into three directions roughly. First one is the perfecting of Bilbo system on corpus level 1 thanks to the fully noted labeling error reports and the correctness of wrong annotation by a specialist in the domain. One of the frequent error patterns of Bilbo is the confusion between author name and place name. This confusion can be eliminated using the proper noun dictionaries already possessed by Cléo. Second, we continue to construct new CRF models on the corpus level 2. Since this corpus is more difficult to be adapted to a learning system, there would be many interesting processes to be applied for the preparation of an appropriate learning data. The similar operation is required also on the corpus 3, on which we just started the manual annotation. As the third corpus is intended for the contextual references, we are thinking about using latent probabilistic models to extract internal structure of the articles that can be used for verifying reference parts in the body of text. Third direction is the usability of auto-labeling results from Bilbo system into other platforms. The identified reference fields can be used to enrich other related external documents with any types of platform such as blog, sns or online books.

7. ACKNOWLEDGMENTS

We thank Google for the Google Grant for Digital Humanities which supports this research project.

8. REFERENCES

[1] M.-Y. Day, T.-H. Tsai, C.-L. Sung, C.-W. Lee, S.-H. Wu, C.-S. Ong, and W.-L. Hsu. A knowledge-based approach to citation extraction, information reuse and integration. In *Proceedings of IRI -2005 IEEE International Conference*, pages 50–55, 2005.

[2] H. Han, W. Xu, H. Zha, and C. L. Giles. A hierarchical naive bayes mixture model for name disambiguation in author citations. In *ACM SYMPOSIUM ON APPLIED COMPUTING*, pages 1065–1069. ACM, 2005.

[3] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[4] F. Peng and A. McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42:963–979, July 2006.

[5] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[6] C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 2011. To appear.

[7] V. I. Torvik and N. R. Smalheiser. Author name disambiguation in medline. *ACM Trans. Knowl. Discov. Data*, 3:11:1–11:29, July 2009.

[8] R. Volz, J. Kleb, and W. Mueller. Towards ontology-based disambiguation of geographical identifiers. In *Proceedings of the WWW2007 Workshop*, 2007.