Bibliographic Attributes Extraction with Layer-upon-Layer Tagging

Wei Wei
School of ICT
Royal Institute of Technology
KTH, Sweden
weiwe@kth.se

Irwin King
Dept. of Comp. Sci. & Eng.
The Chinese University of Hong Kong
Shatin, Hong Kong
king@cse.cuhk.edu.hk

Jimmy Ho-Man Lee
Dept. of Comp. Sci. & Eng.
The Chinese University of Hong Kong
Shatin, Hong Kong
jlee@cse.cuhk.edu.hk

Abstract

Bibliographic attributes extraction is an important research topic for digital libraries. In this paper we propose a rule-based method for bibliographic attributes extraction with Layer-upon-Layer Tagging (LLT). The method analyzes bibliographic attributes' appearances and punctuations to perform format and semantic taggings on two defined parsing layers. The method also resolves to specifically constructed lexicons to achieve high accuracy of semantic tagging. In the experimental evaluation on 1,000 reference strings, the accuracy of author tagging reaches to 96.8% and the accuracy of whole reference tagging is 82.9%. The experimental results demonstrate that the proposed LLT method can tag bibliographic attributes in reference strings with high degree of accuracy.

1. Introduction

Bibliographic information has been significantly utilized in digital libraries. With the rapid growth of digital information resources available, it becomes quite useful and important to be able to extract bibliographic information automatically. A method for automatically and accurately generating structured machine-understandable data from unstructured reference strings is in urgent demand.

Because of the heterogeneity of the reference structure, a universal method to tackle the problem of bibliographic attributes extraction still faces some challenges. The problem was usually solved as the Name Entity Recognition (NER) problem. Methods based on statistical models such as Hidden Markov Models [6, 7], Maximum Entropy Models [9], Conditional Random Field Models [8] are proposed for solving NER problems. Doan [5] proposed a multistrategy learning approach to match schemas of data sources. However, these methods are not dedicated to extracting bibliographic attributes from reference strings. Furthermore, the models in these methods need to be well trained. Takasu [10] proposed an Extended Hidden Markov Model for extracting bibliographic attributes from reference strings captured using OCR, but the proposed model still needs well-prepared training data. Chowdhury [3] mentioned that template mining can be used for information extraction from digital documents and also pointed out that in order to facilitate template mining, standardization in the presentation style and layout of information within digital documents has to be ensured. Ding [4] produced four templates for information extraction from citing and cited articles. However, the result by using template mining still heavily depends on the style and layout of the digital documents. Besagni [2] proposed a method based on part-of-speech tagging for bibliographic reference segmentation. The method proposed did not fully utilize the rules in reference strings and the result is less than satisfactory.

In this paper we propose a rule-based method for bibliographic attributes extraction with Layer-upon-Layer Tagging (LLT). The method analyzes the difficulties and rules of bibliographic attributes extraction to tackle the problem by performing format and semantic taggings on two defined parsing layers. Consider the following two references from two published scientific papers.

- Example 1: "Template mining for information extraction from digital documents", G. Chowdhury, Library Trends, vol. 48, 1999.
- Example 2: Chowdhury, G.G. Template mining for information extraction from digital documents. Library Trends. 48(1), pp.182-208, 1999

The two references both refer to the same article, but the details of the two references' expressions are different. We briefly summarize typical difficulties in solving the problem as follows:

- **The attribute fields delimiters may vary among different reference strings.** For example, commas are used as delimiters between attribute fields in Example 1 while full stop points are used as delimiters in Example 2.
- **Delimiters for attribute fields may also be used within an attribute field.** For example, commas are used between the author's given name and family name in Example 2 while no commas are used for this purpose in Example 1.

Table 1: Format Tags

Tag	Meaning	Example	Tag	Meaning	Example
IW	Word with Capital Initial	Word	LW	Minuscule Word	word
CC	Single Capital Letter	W	UW	Sequential Capital Letters	WORD
DY	Digital Year Number	2006	DN	Digital Number(not year)	312
UY	Capital Letters with Year Number	ICDAR-07	WW	a Hyphen between Two Words	web-based
ON	Ordinal Number	21st	UU	a Hyphen between Two Letters	K-F
DD	a Hyphen between Two Digital Numbers	123-456	DB	Only Digital number and Brackets	53(5)
OT	Others	oThers			

- **The appearances of attributes are not in a definite order.** For example, title appears before author's name in Example 1, but in Example 2 title appears after author's name.
- **Some attributes do not always appear.** For example, a page number field appears in Example 2, but the same is ignored in Example 1.
- **Some reference strings even contain input errors.** For example, reference strings produced by OCR may contain some input errors.

Although the references' expressions vary greatly, we can still discover some rules as follows which is well utilized in our proposed method.

- **Every attribute fields are continuous and not interrupted by each other.**
- **Each attribute field appears only once if it appears at all in a reference string.**
- **Although authors' names cannot be listed exhaustively, common given names and family names frequently exist in a lexicon.**
- **If a substring does not contain any punctuations such as commas, periods, question marks, quotation marks, excalmatory marks, etc., the substring belongs to the same attribute field and cannot be segmented.**
- **The string of publication attribute field is likely to contain particular words such as proceedings, journal, ACM, IEEE, etc.**

The rest of this paper is organized as follows. In Section 2, we briefly introduce preliminaries. In Section 3, we formulate the LLT method in details. In Section 4, we present a simple experiment to evaluate our proposal. In Section 5, we conclude this paper.

2 Preliminaries

In the LLT method, a reference is parsed and analyzed on two different layers. Predefined tags are assigned to substrings out of different layer parsing. During the process of tagging, we also make use of lexicons. In this section we present these preliminary knowledge.

2.1 Parsing layers in LLT

Upperlayer reference parsing. Upperlayer reference parsing is to parse a reference string into coarse-grained substrings with delimiters in punctuation set D_1 of {‘ , ’ ‘ . ’ ‘ ! ’ ‘ ? ’ ‘ ” ’}. The coarse-grained substrings obtained by upperlayer parsing are called **block** substrings. Here is an example of upperlayer reference parsing. In the example, every block substrings are enclosed by "{}".

> {Chowdhury,} {G.} {G.} {Template mining for information extraction from digital documents.} {Library Trends.} {48(1),} {pp.182-208,} {1999.}

Underlayer reference parsing. Underlayer reference parsing is to parse a reference string into fine-grained substrings with delimiters in set D_2 where D_2 is a combination of the punctuation set {‘ , ’ ‘ . ’ ‘ ! ’ ‘ ? ’ ‘ ” ’ ‘ : ’ ‘ (’ ‘) ’} and the blank set which is a set of characters of the space blank and the tab blank. The fine-grained substrings obtained by underlayer parsing are called **patch** substrings. Here is an example of underlayer reference parsing. In the example, every patch substrings are enclosed by "[]";

> [Chowdhury,] [G.] [G.] [Template] [mining] [for] [information] [extraction] [from] [digital] [documents.] [Library] [Trends.] [48] [(1),] [pp.] [182-208,] [1999.]

2.2 Tags used in LLT

Two kinds of tags are defined in the LLT method. **Format tags** are used to denote word appearance as defined in Table 1. **Semantic tags** are used to denote bibliographic semantics as defined in Table 2.

Table 2: Semantic Tags

Tag	Meaning	Tag	Meaning
AU	Author	TT	Title
CJ	Conference or Journal	LC	Location
PG	Page	YR	Year
MT	Month	WK	Week
VO	Volume	UN	Unknown

2.3 Lexicons used in LLT

In the LLT method, several lexicons are specifically constructed for semantic tagging. They are People's Grain Name lexicon, Location Name lexicon, and Publication Name lexicon.

The People's Grain Name (PGN) lexicon is used for author tagging. People's grain name is a minimum grain of a person's name. For example, if a person's name is "James Van Der Beek", the grain names are "James", "Van", "Der" and "Beek". We build the PGN lexicon in the sense that although people's full names cannot be enumerated exhaustively, their given names and family names are mostly in a set of common grain names. In this way, in order to identify a strange name saying "Bill Hanks" we do not have to have the exact name in the PGN lexicon, since we have grain names like "Bill" and "Hanks" which can be easily collected into our lexicon due to two famous existing name "Bill Gates" and "Tom Hanks". In the process of PGN lexicon building, we use author names in **DBLP** to generate the lexicon.
The Location Name (LN) lexicon contains data from an online geographical database [1] which includes all countries and main cities names in the world. The LN lexicon is used for location tagging.
The Publication Name (PN) lexicon contains conferences and journals information in **DBLP**. The PN lexicon is used for publication tagging.

2.4 Particulars in a reference

In a reference, some particular bibliographic attributes often have particular formats. In the LLT method, we fully utilize these particulars and summarize them as follows:

- **Page Number** as a bibliographic attribute in a reference string likely begins with some particular words such as "pp.", "pages", "p.", etc. In the LLT method, these particular words for page numbers are called **PageClews** words.
- **Volume Number** sometimes appears in references for journal papers. The characters within issue and volume substring may only contain characters of '(', ')', '-' and digital numbers. Some particular words, such as "vol", sometimes may appear before volume numbers. In the LLT method, these particular words for volume numbers are called **VolClews** words.
- **Publications Names** likely contain particular words such as "Proceedings", "IEEE", "Transactions", etc. In the LLT method, these particular words for publications are called **PubClews** words.
- **Year Number** as a bibliographic attribute in a reference string is always a number between 1900 and 2007.
- **Month and Week** can easily be tagged and extracted since their forms can be enumerated exhaustively.

3. Layer-upon-Layer Tagging

3.1 Preprocessing

Before parsing and analyzing, each reference string is preprocessed with URL format checking. The purpose of URL format checking is to extract and remove URL substrings from a reference string.

3.2 Parsing on two layers

The first step of LLT process is to parse a reference on different layers. An example as follows is a reference parsed into both block substrings enclosed by "{}" and patch substrings enclosed by "[]". In the example, we can find that a block substring may contain several inside patch substrings. It is obvious that every patch substrings belonging to the same block substring should belong to the same bibliographic attribute.

{[Chowdhury,]} {[G.]} {[G.]} {[Template] [mining] [for] [information] [extraction] [from] [digital] [documents.]} {[Library] [Trends.]} {[48(1),]} {[pp.] [182-208,]} {[1999.]}

3.3 Tagging on two layers

Tagging process involves three steps on two different layers. The first step is underlayer format tagging. The second step is underlayer semantic tagging. The final step is upperlayer semantic tagging.

3.3.1 Underlayer format tagging

In the process of underlayer format tagging, predefined format tags are assigned to every patch substrings. Besides, several other properties of patch substrings are also made clear during this process. The properties of a patch substring are summarized as follows:

- **content**: pure content without any punctuations and delimiters of the patch substring;
- **block**: which block substring the patch substring belongs to;
- **format**: will be assigned with a format tag according to the word appearance of its content;
- **semantic**: will be assigned with a semantic tag;

3.3.2 Underlayer semantic tagging

Underlayer semantic tagging is to tag each patch substring with a predefined semantic tag. It involves tagging all the bibliographic attributes in a reference. The process is described in Algorithm 1.

Algorithm 1 Underlayer Semantic Tagging

```
IF patch.format=="IW" AND patch.content in PGN lexicon:
        patch.semantic = "AU";
ELSE IF patch.format=="DD" OR patch.content is a PageClews word:
        patch.semantic = "PG";
ELSE IF patch.format== "DY":
        patch.semantic = "YR";
ELSE IF patch.content is a month name:
        patch.semantic = "MT";
ELSE IF patch.content is a week name:
        patch.semantic = "WK";
ELSE IF patch.format=="DB" OR patch.content is a VolClews word:
        patch.semantic = "VO";
ELSE IF patch.content is a PubClews word:
        patch.semantic = "CJ";
ELSE IF patch.content in LN lexicon:
        patch.semantic = "LC";
ELSE:
        patch.semantic = "UN";
```

3.3.3 Upperlayer semantic tagging

In the process of upperlayer semantic tagging, a predefined semantic tag is assigned to a block substring. All the properties with a block substring are:

- **semantic**: will be assigned with a semantic tag and its initialized value is "UN";
- **patchList**: a list of patch substrings in the block substring;
- **total**: total number of patch substrings in the block substring;

The process of upperlayer semantic tagging is divided into three parts: trivial attributes tagging, author tagging and title and publication tagging.

Trivial attributes tagging. Some trivial bibliographic attributes including year, month, week, location, page, etc., are straightforward to be tagged. From the survey of 300 randomly collected references, we find that the total numbers of patch substrings in the block substrings for these trivial bibliographic attributes are equal to or less than 3. Suitable semantic tags can be assigned to corresponding block substrings by referring to their inside patch substrings' semantic tags. An algorithm of upperlayer semantic tagging for trivial bibliographic attributes is shown in Algorithm 2.

Author tagging. In order to tag all the author names in a reference, we have to first know all the potential author block substrings. We already have a PGN lexicon to identify whether a word can be a person's name. A potential mistake made by the PGN lexicon is that sometimes a person's name appears as a word in the title but is mistagged as "AU". From the survey of 300 randomly collected references, we find that 99.8% block substrings for author names contain at most 4 patch substrings. Therefore, a potential author block substring should at least contains one patch substring with semantic tag "AU" and its total number of patch substrings is less than or equal to 4. This rule effectively avoids mistagging persons' names in a title as "AU". The algorithm to tag all the potential author block substrings is shown in Algorithm 3. After finding all the potential author block substrings, the longest continuous potential author block substrings will be tagged with "AU". In the process of finding the longest continuous potential author block substrings, some block substrings such as "{[G.]}" will also be tagged with "AU", if its predecessor or successor block substring can be tagged with "AU".

Title and publication tagging. Usually, title block substrings and publication block substrings are the two longest block substrings in a reference. However, title block substrings and publication block substrings are very similar since many words can overlap in the two kinds of substrings. In the LLT method, we not only utilize the PN lexicon but also make use of some inherent rules of bibliographic attributes. One important rule is that publication names may contain **pubClews** words. Another rule we utilize in our algorithm is that title often appears before publication in a reference string. The algorithm is shown in Algorithm 4.

Algorithm 2 Trivial Attributes Tagging

```
FOR all block substrings in a reference:
    IF block.total⩽ 3:
        FOR all patch substrings in a block substring:
            IF patch.semantic=="YR":
              patch.block.semantic="YR";
              break;
            ELSE IF patch.semantic=="MT":
              patch.block.semantic="MT";
              break;
            ELSE IF patch.semantic=="WK":
              patch.block.semantic="WK";
              break;
            ELSE IF patch.semantic=="LC":
              patch.block.semantic="LC";
              break;
            ELSE IF patch.semantic=="PG":
              patch.block.semantic="PG";
              break;
            ELSE:
              patch.block.semantic="UN";
    ELSE:
      block.semantic="UN";
```

Algorithm 3 Potential Authors Tagging

```
Blocks: a sequential list of block substrings in a reference
FOR i TO Blocks.size:
    IF Blocks[i].total⩽ 4:
        FOR j TO Blocks[i].patchList.size:
            IF Blocks[i].patchList[j].semantic=="AU":
                Blocks[i].semantic = "AU";
                break;
```

Algorithm 4 Title and Publication Tagging

```
find the two longest block substrings with semantic tag of "UN";
IF one of two substring can be found in PN lexicon:
        tag the found substring with "CJ";
ELSE IF only one of them contains pubClews words:
        tag the one with pubClews words with "CJ"
        tag the other one with "TT"
ELSE:
        tag the one appearing earlier with "TT"
        tag the other one with "CJ"
```

Table 3: LLT Experimental Results Statistics

Attributes	Author	Title	Publication	Page	Volume	Year	Location	Whole Ref
Correct	**96.8%**	86.3%	88.7%	96.2%	95.4%	98.1%	95.8%	**82.9%**
Incomplete	2.4%	1.2%	1.6%	0.3%	0.01%	0.0%	1.6%	17.1%
Wrong	0.8%	12.5%	9.7%	3.5%	4.5%	1.9%	2.6%	0.0%

4 Experimental Results

The experiment is performed on a data set of 1,000 references randomly collected from 100 published papers. All the test references are first tagged manually and then automatically tagged with the LLT method. We examine the outcomes of the LLT method according to manually tagged references and sum up familiar failures of the LLT method as follows:

- For the author tagging, as the punctuation ":" could be used either as a delimiter between the author attribute and the title attribute or as an inside punctuation of the title, the LLT method fails to differentiate these two types of situations.
- For the title tagging, the LLT method fails to detect the whole title, if the title is made up of more than one sentence.
- For the publication tagging, the LLT method may be confused by the publication attribute and the title attribute, if the name of the publication is neither in the PN lexicon nor contains any **PubClews** words.
- For the tagging of page, volume, and year, the LLT method may be confused, if the page number or the volume number happen to be a year number.
- For the location tagging, the LLT method fails to recognize the location attribute, if the name of the location is not in the LN lexicon.

We further calculate the percentages of tagging accuracies for every bibliographic attributes in Table 3. From the Table 3, we may roughly conclude that the accuracies of most attributes tagging by the proposed LLT method are superior to the results by Besagni's approach [2]. Specifically, the accuracy of author tagging by the LLT method reaches to 96.8%, which is greatly superior to the accuracy of author tagging (90.2%) of Besagni's approach.

5 Conclusions

This paper presents a rule-based method of bibliographic attributes extraction with layer-upon-layer tagging. It is different from traditional approaches based on statistical models. The LLT method is rule-based and fully utilizes the rules in reference strings. It analyzes bibliographic attributes' appearances and punctuations to perform format and semantic tagging on two defined parsing layers. For some particular attributes, such as authors, publications and locations, the LLT method resolves to three specifically constructed lexicons for assistance in the semantic tagging process. We conduct an experimental evaluation on 1,000 reference strings for LLT and achieve the accuracy of author tagging up to 96.8% and accuracy of whole referecnes tagging up to 82.9%. The experimental results demonstrate that the proposed LLT method can tag bibliographic attributes in reference strings with high degree of accuracy. We believe the accuracy of LLT can be improved if we merge statistical approaches in the process of semantic tagging.

References

[1] Geonames.org: an online geographical database. http://www.geonames.org/.

[2] D. Besagni, A. Belaïd, and N. Benet. A segmentation method for bibliographic references by contextual tagging of fields. In *ICDAR*, pages 384–388, 2003.

[3] G. G. Chowdhury. Template mining for information extraction from digital documents. *Library Trends*, 48(1), 1999.

[4] Y. Ding, G. Chowdhury, and S. Foo. Template mining for the extraction of citation from digital documents. In *Second Asian Digital Libraries Conference*, 1999.

[5] A. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources. A multistrategy approach. *Machine Learning*, pages 279–301, 2003.

[6] T. Grenager, D. Klein, and C. D. Manning. Unsupervised learning of field segmentation models for information extraction. In *ACL*, 2005.

[7] M. Inoue and N. Ueda. Exploitation of unlabeled sequences in hidden markov models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(12):1570–1581, Dec. 2003.

[8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of 18th Int. Conf. on Machine Learning*, 2001.

[9] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598, 2000.

[10] A. Takasu and K. Aihara. Bibliographic component extraction from references based on a text recognition error model. *Systems and Computers in Japan*, 36(7):13–22, 2005.