FLUX-CiM: Flexible Unsupervised Extraction of Citation Metadata

Eli Cortez[1], **Filipe Mesquita**[1], **Altigran S. da Silva**[1]
Edleno Moura[1], **Marcos André Gonçalves**[2]

[1] Universidade Federal do Amazonas – Departamento de Ciência da Computação

[2]Universidade Federal de Minas Gerais – Departamento de Ciência da Computação

{eccv,fsm,alti,edleno}@dcc.ufam.edu.br, mgoncalv@dcc.ufmg.br

Abstract. *In this paper we propose FLUX-CiM, a tool that extracts components of citations in any given format. Differently from related systems that rely on manually built examples for recognizing the components of a citation, we rely on an existing set of sample metadata records from a given area (e.g., computer science or health sciences). Our tool does not rely on patterns encoding specific delimitators of a particular citation style. It is also unsupervised, in the sense that it does not rely on a learning method that requires a training phase. These features assign to our tool a high degree of automation and flexibility.*

1. Introduction

Citation management is a central aspect of modern digital libraries. Citations serve, for example, as a fundamental evidence of the impact or significance of particular scientific articles, and therefore of the research they report. Evaluation of individual's performances for promotions and grants may use citations as evidence to evaluate competence and the impact of a researcher's work. Citation management in a digital library involves aspects such as: (i) data cleaning to correct mistakes, such as assignment of improper authorship or splitting of a researcher's production due to the use of multiple names in publications; and (ii) removal of duplicates, mainly after data integration or data input tasks. Most of the techniques to perform these tasks rely on the assumption that we can correctly identify main components within a citation, such as authors' names, title, publication venue, year, pages, etc. This, although is not an easy task due to a variety of reasons such as [Lee et al. 2007]: data entry errors, various citation formats, lack of (the enforcement of) a standard, imperfect citation gathering software, common author names, abbreviations of publication venues and large-scale citation data.

In this paper we present a tool for extracting components of citations in any given format, which implements a knowledge-base (KB) approach presented in [Cortez et al. 2007]. Differently from similar systems such as [Embley et al. 1999, Day et al. 2005] that rely on manually built knowledge-bases for recognizing the components of a citation, in our case, such a KB is automatically constructed from an existing set of sample metadata records from a given area (e.g., computer science or health sciences). Such sample metadata records are very easy to obtain nowadays, for instance, by collecting them directly from the web or by harvesting open archives. Therefore, FLUX-CiM is unsupervised, since it does not rely on a learning method that requires a, sometimes very expensive, training phase. It can be applied in any bibliographic citation field as long as a knowledge base can be constructed, which is easily done with relatively little effort as we shall see.

The extraction process in our tool is based on: (1) estimating the probability of a given term found on a citation to occur in a value of a given citation field according to the information encoded in the KB, and (2) the use of generic structural properties of bibliographic citations. This means that our approach does not rely on patterns encoding specific delimitators of a particular citation style. This assigns to our tool a high degree of automation and flexibility, as demonstrated by experiments we have reported here. A demo of the FLUX-CiM tool is available in `http://vitoria.dcc.ufam.edu.br/flux/`.

This paper is organized as follows. Section 2 presents an overview of the FLUX-CiM tool. Section 3 presents in details the method implemented in the tool. Section 4 shows an experiment comparing FLUX-CiM with CRF, a state-of-art extraction model.

2. Overview

(a) Architecture (b) An example of interface

Figure 1. Overview of FLUX-CiM

The FLUX-CiM tool works as illustrated in Figure 1(a) and an example of interface is shown in Figure 1(b). FLUX-CiM takes as input a set of *citation strings*, often using simple format converters that extract text from files in PDF and other popular formats. A citation string is a text portion encompassing a complete citation from the list of citations in a paper file. Our tool recognizes components of these citations using a knowledge base and *potential delimiter characters* found in citation, as discussed below.

A knowledge base is a set of pairs $KB = \{\langle m_1, O_1\rangle, \ldots, \langle m_n, O_n\rangle\}$ in which each m_i is a distinct bibliographic metadata field, and O_i is a set of strings $\{o_{i,1}, \ldots, o_{i,n_i}\}$ called *occurrences*. Intuitively, O_i is set of typical values for field m_i. In our implementation, the knowledge base is represented as an inverted index composed by the terms found in the occurrences. In Figure 2 we present a very simple example of a knowledge base, which includes only two metadata fields: *Author* and *Title*.

A *potential delimiter character*, or *p-delimiter*, is any character other than words (`a`–`z`) and numbers (`0`–`9`). We notice that we do not assume p-delimiters as field delimiters intrinsically. Instead, as explained below, we keep track of them to verify if they indeed are used as delimiters in the citation string being processed.

$KB = \{\quad \langle Author, O_{Author} \rangle, \langle Title, O_{Title} \rangle\}$
$O_{Author} = \{\quad$ "J. K. Rowling","Galadriel Waters", "Beatrix Potter"$\}$

$O_{Title} = \{\quad$ "Harry Potter and the Half-Blood Prince",
"A Guide to Harry Potter", "Petter Rabbit's Halloween"$\}$

Figure 2. A sample knowledge base.

3. The FLUX-CiM Method

The FLUX-CiM method consists of four steps, as illustrated in Figure 3. In the *blocking* step, citation strings are split in syntactic units called *blocks*. In the *matching* step, we attempt to associate a citation metadata field to each block based on the information available on the knowledge base. After this, in the *binding* step, blocks left unassociated in the previous step are further analyzed for associations based on their relative position on the citation string. Finally, the *joining* step composes metadata values by joining contiguous blocks associated to the same field.

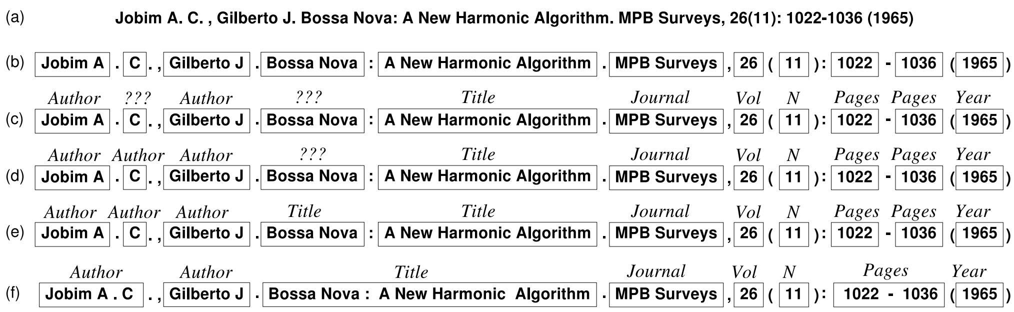


Figure 3. An sample citation string (a) and the extraction steps: blocking (b), matching (c), binding (d and e), and joining (f).

3.1. Blocking

The first step in our extraction method consists of splitting every citation string into substrings we call *blocks*. Let p_l and p_r be p-delimiters and C be a citation string. A block b is an string containing no p-delimiters that occurs in a sequence $p_l b p_r$, or $b p_r$ where b is a prefix of C, or $p_l b$ where b is a suffix of C. In a same citation string, there could be more than one block that will be associated to a same field. In Figure 3(b) the blocks identified for our example citation string are marked with rectangles. The rationale behind the idea of identifying blocks is the observation that, in general, in a citation string, every field value is bounded by a p-delimiter, but not all p-delimiters bound a field.

3.2. Matching

The matching step consists of associating each block with a bibliographic metadata field. To accomplish this, we match each block against the occurrences composing the knowledge base and evaluate to which field the block is more likely to belong to. To account for this, we use for the matching a function we call *FF* (Field Frequency), which is an adaptation of the AF function proposed in [Mesquita et al. 2007]. The FF function is defined below.

$$FF(b, m_i) = \frac{\sum\limits_{t \in T(m_i) \cap T(b)} \textit{fitness}(t, m_i)}{|T(b)|} \quad (1)$$

where $T(m_i)$ is the set of all terms found on the occurrences of metadata field m_i, and $T(b)$ is the set of terms found in block b.

The FF function estimates the probability of b being a part of an occurrence of m_i, by evaluating how typical the terms in b are in the occurrences of this field according to the knowledge base. For this, we define a *fitness measure* which attempts to measure how typical a given term is for each field where it occurs. For instance, in the occurrences of Figure 2, the term Potter is more typical in field *Title* than in field *Author*.

The fitness measure is computed by the following formula:

$$fitness(t, m_i) = \frac{f(t, m_i)}{N(t)} \times \frac{f(t, m_i)}{f_{max}(m_i)} \quad (2)$$

where $f(t, m_i)$ is the number of occurrences $o_{i,k} \in O_i$ associated with field m_i in the knowledge base which contain the term t, $f_{max}(m_i)$ is the highest frequency of any term among the occurrences $o_{i,k} \in O_i$, and $N(t)$ is the total number of occurrences of term t in the knowledge base.

The first fraction in Equation 2 expresses the probability of term t be part of an occurrence of m_i in the knowledge base. Such probability would be suitable for our purposes with all m_i had the same number of occurrences in the knowledge based. As this not true in general, fields with more occurrences would tend to have higher probability values. Therefore, we add the second fraction, as a normalization factor to avoid this problem. This fraction gives the frequency of t in occurrences of m_i normalized by maximum frequency of a term in occurrences of m_i.

Thus, for each block b in the citation string, we calculate $FF(m_i, b)$, for every field m_i in the knowledge base. Finally, b is associated to the field which gives the maximum FF value. However a block is left *unmatched* if any of its terms is found in KB. In Figure 3(c) unmatched blocks are labeled with *???* and matched blocks are labeled with the names of their corresponding fields.

3.3. Binding

The binding step associates remaining unmatched blocks with fields. There are three distinct cases we consider: *homogeneous neighborhood*, *partial neighborhood* and *heterogeneous neighborhood*. For each of these cases, we detail below the specific binding strategy adopted.

Homogeneous Neighborhood

Let l and r be matched blocks associated to a same field m. Suppose these blocks occur in a sequence $l, p_0, u_1, p_1, \ldots, u_n, p_n, r$, in which each u_i is a unmatched block and each p_i is p-delimiters. In this case, all u_i will be associated to m. An example of homogeneous neighborhood is illustrated in Figure 3(c), where the block containing the term "C" is associated to *Author* in Figure 3(d) since both of its neighbors are associated to this field.

Partial Neighborhood

Let b be a matched block associated to field m. Suppose this block occur in a sequence $I = u_1, p_1, \ldots, u_n, p_n, b$ or in a sequence $F = b, p_0, u_1, p_1, \ldots, u_n$, in which each u_i is a unmatched block and each p_i is a p-delimiter. In this case, all u_i will be associated to m. Notice that in I, blocks u_i begin the citation string, while in F, blocks u_i end the citation string.

Heterogeneous Neighborhood

Consider the example in Figure 3(c), where we must decide whether the block containing "Bossa Nova" should be associated to *Author*, as the block on the left, or to *Title* as the block on the right. In such situations, our method resorts to the available p-delimiters surrounding the unmatched blocks, and verifies if which of them are indeed field delimiters. This verification is carried out based on the results of the matching step for a set of citations, where several blocks are labeled with their corresponding field. For instance, in Figure 3, because "." is likely to be a delimiter between *Author* and *Title* and ":" is likely to be a character occurring in values of *Title*, we would choose to associate "Bossa Nova" to *Title* rather than to *Author*. These ideas are elaborated in the following.

Consider the sequence $l, p_0, u_1, p_1, \ldots, u_n, p_n, r$, where l and r are matched blocks associated to distinct fields m_l and m_r, respectively, u_i are unmatched blocks and p_i are p-delimiters. Our problem is to determine, for each u_i, whether it will be associated to m_l or to m_r. First of all, we consider that only one p-delimiter p_i is indeed a field delimiter. Based on this, once we find that some p_i is a field delimiter, then we associate all unmatched blocks u_j $(0 < j \leq i)$ to m_l, i.e., same field as the block on the left, and we associate all u_k $(i > k \geq n)$ to m_r, i.e., same field as the block on the right.

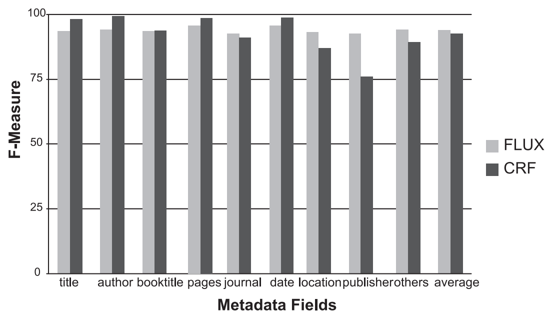
Consider a function $D(p_k, m_l, m_r)$ that estimates the probability of a p-delimiter p_k being a field delimiter between blocks associated to fields m_l and m_r, respectively. Thus, the problem of binding the sequence of unmatched blocks within a heterogeneous neighborhood is solved by calculating $D(p_k, m_l, m_r)$ for each p-delimiter p_k in the sequence. The function D is precisely defined in [Cortez et al. 2007]. The field delimiter is selected as the one for which this equation gives the largest value. In Figure 3(e), for instance, the block containing the term "Bossa Nova" is associated to *Title*, since D(":", *Title*, *Author*) $<$ D(".", *Title*, *Author*).

3.4. Joining

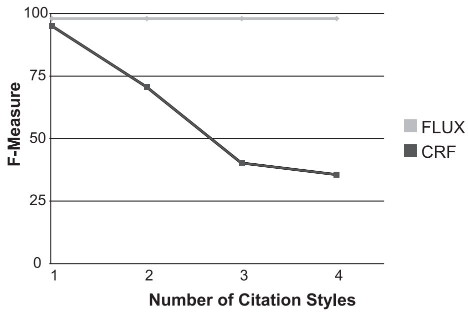
The last step in our extraction method is joining together contiguous blocks associated to a same field to form the values of that field. For most of the cases, this step is straightforward to accomplish; however, joining blocks associated to the *Author* field requires a more careful procedure, since there may be several values of this field on each citation string. In this case, we join every contiguous blocks $b_i p b_j$, except when p is an implicit delimiter for separating the values of *Author*. We define a set of p-delimiters as *value delimiters* by comparing the average length of values surrounded by them in the every citation string with the average length of *Author* values in the KB, as detailed in [Cortez et al. 2007]. In Figure 3(f) we show the *Author* values obtained with delimiter ",".

4. Experimental results and conclusions

We have experimented our method, and compared it with the state-of-art in the literature, Conditional Random Fields (CRF). As a result, the extraction quality obtained by FLUX-CiM, even without user intervention, reached F-Measure levels above 92% (almost 3% higher in average than CRF). These experimental results were obteined on three distinct citation datasets, including CORA, the one used in the original CRF paper, as shown in Figure 4(a). In another experiment we carried out, results obtained showed that our method is capable of dealing with several distinct citation styles without compromising the extraction quality, a feature not present in CRF whose extraction quality degrades with the number of distinct citation styles used, as illustrated in Figure 4(b). Some results of this research were published at [Cortez et al. 2007]. The results obtained in these experiments in comparison with the state-of-art research, lead us to regard our method as the best cost effective method for metadata citation extraction in the literature.



(a) CORA



(b) Health Sciences

Figure 4. Comparative evaluation between FLUX-CiM and CRF for several fields (a) and when dealing with several citation styles (b).

References

Cortez, E., da Silva, A., Gonçalves, M., Mesquita, F., and de Moura, E. (2007). FLUX-CIM: flexible unsupervised extraction of citation metadata. In *Proceedings of the 2007 conference on Digital libraries*, pages 215–224. ACM Press New York, NY, USA.

Day, M.-Y., Tsai, T.-H., Sung, C.-L., Lee, C.-W., Wu, S.-H., Ong, C.-S., and Hsu, W.-L. (2005). A knowledge-based approach to citation extraction. In *IRI '05: Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration*, pages 50–55, New York, NY, USA. IEEE Systems, Man, and Cybernetics Society.

Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Lonsdale, D. W., Ng, Y.-K., and Smith, R. D. (1999). Conceptual-model-based data extraction from multiple-record web pages. *Data Knowl. Eng.*, 31(3):227–251.

Lee, D., Kang, J., Mitra, P., Giles, C. L., and On, B.-W. (2007). Are your citations clean? new scenarios and challenges in maintaining digital libraries. To appear in Communications of the ACM.

Mesquita, F., da Silva, A. S., de Moura, E. S., Calado, P., and Laender, A. H. F. (2007). Labrador: Efficiently publishing relational databases on the web by using keyword-based query interfaces. *Inf. Process. Manage.*, 43(4):983–1004.