A Flexible Approach for Extracting Metadata From Bibliographic Citations

Eli Cortez and Altigran S. da Silva
Department of Computer Science, Federal University of Amazonas, Av. Gen. Rodrigo Otávio, 3000, CEP 69077-000, Manaus-AM, Brazil. E-mail: {eccv, alti}@dcc.ufam.edu.br

Marcos André Gonçalves
Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte-MG, Brazil. E-mail: mgolcalv@dcc.ufmg.br

Filipe Mesquita and Edleno S. de Moura
Department of Computer Science, Federal University of Amazonas, Manaus-AM, Brazil. E-mail: {fsm, edleno}@dcc.ufam.edu.br

In this article we present FLUX-CiM, a novel method for extracting components (e.g., author names, article titles, venues, page numbers) from bibliographic citations. Our method does not rely on patterns encoding specific delimiters used in a particular citation style. This feature yields a high degree of automation and flexibility, and allows FLUX-CiM to extract from citations in any given format. Differently from previous methods that are based on models learned from user-driven training, our method relies on a knowledge base automatically constructed from an existing set of sample metadata records from a given field (e.g., computer science, health sciences, social sciences, etc.). These records are usually available on the Web or other public data repositories. To demonstrate the effectiveness and applicability of our proposed method, we present a series of experiments in which we apply it to extract bibliographic data from citations in articles of different fields. Results of these experiments exhibit precision and recall levels above 94% for all fields, and perfect extraction for the large majority of citations tested. In addition, in a comparison against a state-of-the-art information-extraction method, ours produced superior results without the training phase required by that method. Finally, we present a strategy for using bibliographic data resulting from the extraction process with FLUX-CiM to automatically update and expand the knowledge base of a given domain. We show that this strategy can be used to achieve good extraction results even if only a very small initial sample of bibliographic records is available for building the knowledge base.

Received September 29, 2008; revised January 13, 2009; accepted January 13, 2009

• Published online 25 February 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.21049

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 an individual's performances for promotions and grants may use citations as evidence to evaluate competence and the impact of a researcher's work. Citations also have been used as an auxiliary evidence in information retrieval tasks such as automatic document classification presented in Calado et al. (2006) and Couto et al. (2006), indexing and ranking proposed by Lawrence, Giles, and Bollacker (1999), and quality assessment as shown in Gonçalves, Moreira, Fox, and Watson (2007). Bibliographic measures that rely on citations have served as inspiration for modern Web link analysis algorithms such as PageRank, presented in Brin and Page (1998). Citations in a broader sense[1] are the basis of important projects such as the Digital Bibliography & Library Project (DBLP; http://www.informatik.uni-trier.de/~ley/db) and the Computer Science Bibliography (http://liinwww.ira.uka.de/bibliography).

Citation management in a digital library involves aspects such as (a) 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 (b) 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 the

[1] Here interpreted as a set of bibliographic information (e.g., author name, title, publication venue, or year) that is pertinent to a particular article.

main components within a citation, such as authors' names, title, publication venue, year, pages, and so on. This is not an easy task due to a variety of reasons, such as those identified by Lee, Kang, Mitra, Giles, and On (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.

We present *FLUX-CiM* (*Flexible Unsupervised Extraction-Citation Metada*), a method to help extract the correct components of citations in any given format. Differently from related approaches that were presented in Embley et al. (1999), Day et al. (2005), and Peng and McCallum (2006) that rely on manually built training data for recognizing the components of a citation, in our case, our method relies on a knowledge base automatically constructed from an existing set of sample metadata records from a given area. Such sample metadata records are very easy to obtain (e.g., collected directly from the Web or harvested from open archives) (Open Archives Initiative, 2005). In brief, our extraction method is based on (a) estimating the probability of a given term found on a citation to occur as a value of a given citation field according to the information encoded in the knowledge base and (b) the use of generic structural properties of bibliographic citations (e.g., the use of punctuation signs to delimit fields). This means that our approach does not rely on patterns encoding specific delimiters of a particular citation style. This gives to our method a high degree of automation and flexibility, as demonstrated by experiments we have conducted and report here.

Preliminary results with our work on FLUX-CiM were previously presented in Cortez, da Silva, Gonçalves, Mesquita, and de Moura (2007), and here, we present a number of extensions to this work, including the following.

We report results of experiments with our method to extract information from citations in three different domains. In the Computer Science area, we used data from the CORA (http://www.cs.umass.edu/∼mccallum/data/cora-ie.tar.gz) that was used in McCallum (2006), in the Health Sciences area, data from several journal articles sponsored by the U.S. National Institutes of Health (NIH), and in the Social Sciences area, we used data from several journal articles sponsored by the Scielo Digital Library (http://www.scielo.org/). To build the knowledge base, we used in the Computer Science case data from CORA itself that was not included in the experimental evaluation. In the Health Sciences and Social Sciences cases, we used metadata records from PubMed Central (PMC; http://www.pubmedcentral.nih.gov/) and the Scielo Digital Library, respectively, both being free digital repositories. Results of these experiments indicated that our method was able to correctly extract, on average, over 94% of the field values present in the citations. In addition, the extraction for more than 82% of the citations was perfect, with all fields correctly extracted.

We also report on experiments carried out to compare our method with *Conditional Random Fields* (CRFs; McCallum, 2006) for solving this problem. CRF is a state-of-the-art approach in information extraction (discussed later). These results corroborate our claims regarding the high quality our method achieves, even without user-assisted training. In particular, FLUX-CiM produced a far-superior performance when the input documents had citations formatted with several different styles.

Finally, we present a strategy for using bibliographic data resulting from the extraction process with FLUX-CiM to automatically update and expand the knowledge base of a given domain. We show that this feedback strategy can be used to achieve good extraction results even if only a few sample bibliographic records are available for building the knowledge base.

This article is organized as follows. First, we cover related work and discuss the background of the concepts used in our approach. After presenting the proposed method in detail, we present our experiments and a comparative study with a state-of-the-art information extraction approach. We then present a strategy for automatically updating and expanding the knowledge base using feedback. We conclude the paper with directions for future work.

Related Work

In past years, several tools, methods, and techniques have been proposed to address the issue of data extraction from textual documents, with a focus on documents available on the Web. A brief survey on this topic is presented in Laender, Ribeiro-Neto, da Silva, and Teixeira (2002b). For dealing with such a problem, several distinct techniques have been deployed, such as HTML structure analysis (Arasu & Garcia-Molina, 2003; Crescenzi, Mecca, & Merialdo, 2001; Liu, Grossman, & Zhai, 2003; Reis, Golgher, Silva, & Laender, 2004), natural language processing (Freitag & McCallum, 2000; Muslea, Minton, & Knoblock, 2001; Soderland, 1999), machine learning (Hsu & Dung, 1998; Kushmerick, 2000), data modeling (Laender, Ribeiro-Neto, & da Silva, 2002a), and ontologies (Embley et al. (1999)).

Most approaches in the literature use training source documents (e.g., Web pages), provided with labeled example values, from which the regularities in the formating surrounding values of interest are learned. In such approaches, the extraction process consists of recognizing and extracting strings within these surroundings, occurring in input documents similar to those provided in the training. However, FLUX-CiM does not rely on formatting features of the input documents (e.g., regularities in value surroundings or page structure) but rather on their content, considering features of the bibliographic fields along with their values. Thus, FLUX-CiM is able to recognize appropriate values to use in the fields regardless of the particular format of the input documents or style used in citation records.

Another content-based approach for data extraction is the one proposed by Embley et al. (1999) on ontology-based data extraction. This approach uses a semantic data model to construct an ontology that describes the data of interest, including relationships, lexical appearances, and context

DOI: 10.1002/asi

keywords. By parsing this ontology, a relational database schema and a constant/keyword recognizer are automatically generated, which are then used to extract data that will populate the database. While most approaches rely on the textual context surrounding the data of interest, the ontology-based approach relies mainly on the expected content of the pages, according to what was anticipated by a prespecified ontology built by a specialist. If the ontology is representative enough, the extraction process is fully automated. In this case, the extraction process is inherently resilient (i.e., it works properly even if the formatting features of the source documents change) and adaptable (i.e., it works for documents from many distinct sources belonging to the same application domain).

In the Digital Library realm, automatic metadata extraction is a rapidly growing related area of research which has been recently gaining much attention. Han et al. (2003) described a Support Vector Machine classification-based method for metadata extraction from the header part of research papers and showed that it outperforms other machine learning methods on the same task. MetaExtract is a system to automatically assign Dublin Core + GEM metadata using extraction through natural language processing techniques applied to educational documents (Yilmazel, Finneran, & Liddy, 2004). Hu, Li, Cao, Meyerzon, and Zheng (2005) focused on title extraction from general documents (e.g., presentations, book chapters, technical papers, brochures, reports, and letters). Paynter (2005) focused on the evaluation of automatic metadata assignment tools and discussed its advantages and limitations. Day et al. (2005) proposed an approach for metadata extraction based on an ontological knowledge representation framework called INFOMAP. This approach, similarly to the one proposed in Embley et al. (1999), requires an ontology to be built, in this case with the help of the Compass editing tool. The authors reported good extraction results considering six different (although fixed) citation patterns for journal articles only.

McCallum (2006) addressed the problem of information extraction of bibliographic data from research papers and proposed the use of CRFs for solving this problem. CRFs, presented in Lafferty, McCallum, and Pereira (2001), are probabilistic models commonly used for extracting information implicitly available on textual sources. They work by assigning labels to segments in the input text. The labeling and the segmentation are based on a model generated from a training process over instances of text manually labeled and segmented. The training aims at capturing several local features (e.g., field sequence, writing style), external lexicon features (e.g., thesauri), and layout features (e.g., punctuation, font style) to be represented in the model. To corroborate their claims regarding the quality of their proposed method, the authors tested it with the CORA dataset, which also was used in our experiments. Currently, CRF constitutes state-of-the-art information extraction due to its flexibility and the quality of the extraction results achieved. Later, we present a comparative study between this method and ours.

The idea of using feedback in information extraction, although not novel, is a trend only recently explored in the literature. It has been previously deployed with CRF in Culotta, Kristjansson, McCallum, and Viola (2006), where the authors presented a study on how to improve extraction models using user feedback. That article also described a framework designed to help users in manually correcting CRF extraction models. We show that our extraction method (FLUX-CiM) allows for using feedback to improve the quality of extraction results in a fully automated fashion (i.e., without user intervention). This is major distinction from the work presented in Culotta et al. (2006), which does require a user to manually guide the feedback process.

The FLUX-CiM Method

In this section, we present the details of our citation metadata extraction method, FLUX-CiM. We provide some concepts and definitions used throughout the discussion, and then discuss each step that comprises our method. First, we discuss the *blocking* step, in which a citation string containing the metadata to be extracted is split in syntactic units called *blocks*. After the blocking step, we discuss the *matching* step, which attempts to associate a citation metadata field with each block based on the information available on the knowledge base. We then discuss the *binding* step, in which blocks left unassociated in the previous step are further analyzed for associations based on their relative position on the citation string. Finally, we discuss the *joining* step, in which blocks are joined to form the values of fields that compose a metadata record.

Basic Concepts

Knowledge Base

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.

The process of building a knowledge base is trivial. Given a set of bibliographic metadata records for a given area, we simply process each record, and for each field, we extract the values as occurrences. Note that this process requires no human effort for selecting some form of "gold standard" records. Indeed, the process is most likely to be automatically carried out by using format conversion. For instance, the knowledge base we built for testing our method over citations in the Computer Science domain came from a set of bibtex entries available in the CORA collection (McCallum, 2006). For this, we simply parse each entry and store field values in our knowledge base. Regarding implementation, in the prototype we used for our experiments, the knowledge base is represented as an inverted index composed by the terms found in the occurrences. In Figure 1, we present a very simple example of a knowledge base which includes only two metadata fields: *Author* and *Title*.

$KB =$	$\langle Author, O_{Author} \rangle$, $\langle Title, O_{Title} \rangle$
$O_{Author} =$	"J.K. Rowling," "Galadriel Waters," "Beatrix Potter"
$O_{Title} =$	"Harry Potter and the Half-Blood Prince" "A Guide to Harry Potter," "Peter Rabbit's Halloween"

FIG. 1. A sample knowledge base.

Citation String

A *citation string* is a text portion encompassing a complete citation from the list of citations in a file. In our method, citation strings are obtained using simple format converters that extract text from files in PDF and other popular formats. In Figure 2a, we present an example of a citation string.

p-delimiters

A *potential delimiter character*, or *p-delimiter*, is any character other than A, . . . , Z, a, . . . , z, or 0, . . . , 9. Note that we do not intrinsically assume p-delimiters as field delimiters. Instead, as explained later, we keep track of them to verify if they indeed are used as delimiters in the citation string being processed.

Method Steps

Blocking. The first step in our extraction method consists of splitting a 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 a string containing no p-delimiters occurring 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 our method, we consider blocks as sets of terms that will compose a value of a certain field. In the same citation string, there could be more than one block that will be associated with a same field. In Figure 2b, 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.

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. For certain terms, this is very easy to accomplish. For instance, the term "procedure" is clearly unrelated to all fields except *Title*. In other cases we have ambiguous terms, and we need to use the occurrences to estimate the degree of ambiguity of terms with respect to the fields on the knowledge base. For instance, consider the simple knowledge base in Figure 1. In these occurrences, the term *Potter* is considered ambiguous since it is found in both occurrences, *Author* and *Title*. On the other hand, the term *Halloween* is typical of the *Title* occurrences, and thus unambiguous.

In the matching phase, textual values (e.g., titles, author names, etc.) are handled using a similarity function we call *Field Frequency* (*FF*), which is an adaptation of the AF function proposed by Mesquita, da Silva, de Moura, Calado, and Laender (2007). The FF function is defined next.

$$FF(b, m_i) = \frac{\sum_{t \in T(m_i) \cap T(b)} fitness(t, m_i)}{|T(b)|} \tag{1}$$

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 $fitness(t, m_i)$ is a function that computes the fitness measure as described later.

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, a *fitness measure* is defined (see Equation 2). Given an ambiguous term, the *fitness* function attempts to measure how typical this term is in each field where it occurs. For instance, in the occurrences of Figure 1, the ambiguous term *Potter* is more typical in field *Title* than it is in field *Author*.

The fitness measure is computed by the following formula:

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

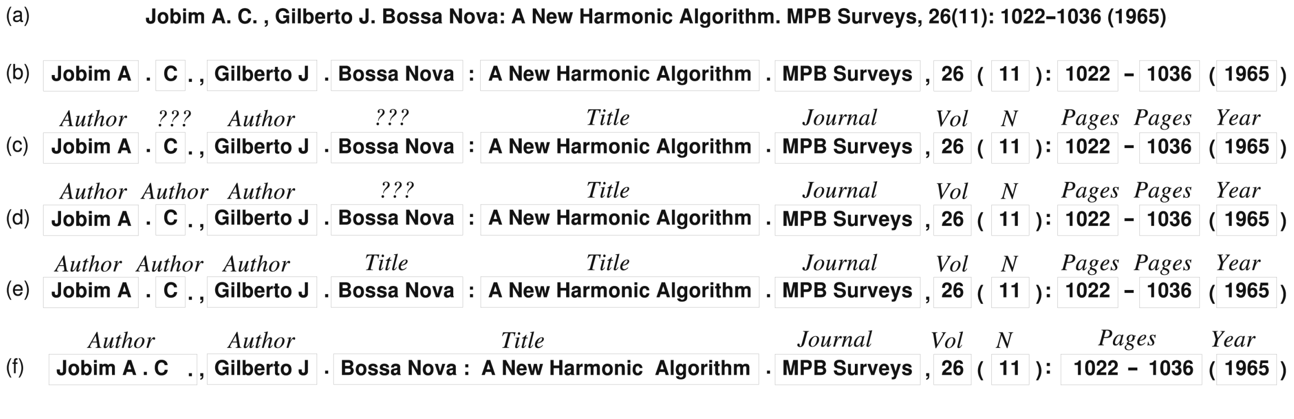

(a)
Jobim A. C. , Gilberto J. Bossa Nova: A New Harmonic Algorithm. MPB Surveys, 26(11): 1022–1036 (1965)
(b)
Jobim A . C . , Gilberto J . Bossa Nova : A New Harmonic Algorithm . MPB Surveys , 26 (11): 1022 – 1036 (1965)
Author ??? Author ??? Title Journal Vol N Pages Pages Year
(c)
Jobim A . C . , Gilberto J . Bossa Nova : A New Harmonic Algorithm . MPB Surveys , 26 (11): 1022 – 1036 (1965)
Author Author Author ??? Title Journal Vol N Pages Pages Year
(d)
Jobim A . C . , Gilberto J . Bossa Nova : A New Harmonic Algorithm . MPB Surveys , 26 (11): 1022 – 1036 (1965)
Author Author Author Title Title Journal Vol N Pages Pages Year
(e)
Jobim A . C . , Gilberto J . Bossa Nova : A New Harmonic Algorithm . MPB Surveys , 26 (11): 1022 – 1036 (1965)
Author Author Title Journal Vol N Pages Year
(f)
Jobim A . C . , Gilberto J . Bossa Nova : A New Harmonic Algorithm . MPB Surveys , 26 (11): 1022 – 1036 (1965)

FIG. 2. A sample citation string (a) and the extraction steps: blocking (b), matching (c), binding (d and e), and joining (f).

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 contains 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 being part of an occurrence of m_i in the knowledge base. Such probability would be suitable for our purposes if all m_i had the same number of occurrences in the knowledge base. Since this is 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 the maximum frequency of a term in occurrences of m_i. Thus, it varies from 0 (*completely infrequent*) to 1 (*most frequent*). This normalization also is useful for making the frequency values comparable among all fields.

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 with the field which gives the maximum FF value.

For the case of numeric values (e.g., page numbers, year, volume, etc.), traditional textual similarity functions do not work properly as it was shown in Agrawal, Chaudhurri, Das, and Gionis (2003), (see Equation 3). Thus, for numeric attributes, we consider a simple, yet effective, approach: We assume that the values in each citation field follow a gaussian distribution. The similarity between the value in the citation and the values of the knowledge base is defined as the mean value of the probability density function. We call this function *Numeric Matching* (*NM*). We normalize this function by the maximum probability density, which is reached when a given value is equal to the mean. Thus, we define the matching score for numeric values as follows:

$$NM(b, m_i) = \frac{1}{|b|} \sum_{v \in b} e^{-\frac{v - \mu}{2\sigma^2}} \qquad (3)$$

where σ and ν are the standard deviation and mean, respectively, of values of m_i.

After the matching step, most of the blocks are associated with one of the fields in the knowledge base. We refer to these blocks as *matched*. However, *unmatched* blocks still may occur; that is, some blocks may remain unassociated with any field after the matching phase. This situation occurs with blocks composed by terms not present in the occurrences of the knowledge base.

In Figure 2c we exemplify an output of the matching step. In this figure, unmatched blocks are labeled with *???* and matched blocks are labeled with the names of their corresponding fields. Cases such as these must be addressed, and this is the task carried out by the binding step explained next.

Binding

In the matching step, several blocks were associated with a field from the knowledge base. Based on this information, the binding step associates remaining unmatched blocks with fields. In Figure 2c, we illustrate two cases of single unmatched blocks (marked with "*???*"). However, in general, there could be a sequence of unmatched blocks that need to be associated with some field. The way we solve this problem depends on the neighborhood of the sequence of unmatched blocks on the citation strings. There are three distinct cases we consider: a *homogeneous neighborhood*, a *partial neighborhood*, and a *heterogeneous neighborhood*. Next, we detail the specific binding strategy adopted for each of these cases.

Homogeneous Neighborhood

Let l and r be matched blocks associated with the 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 a p-delimiter. In this case, all u_i will be associated with m. An example of a homogeneous neighborhood is illustrated in Figure 2c, where the block containing the term "C" is associated with *Author* in Figure 2d since both of its neighbors are associated with this field.

Partial Neighborhood

Let b be a matched block associated with field m. Suppose this block occurs 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 an unmatched block and each p_i is a p-delimiter. In this case, all u_i will be associated with m. Note 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 2c, where we must decide whether the block containing "Bossa Nova" should be associated with *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 (a) they are typically found between contiguous blocks of distinct fields or (b) they are typically found between contiguous blocks of the same field. In the first case, we regard the p-delimiter as being indeed a field delimiter, and thus, the two blocks it separates cannot be associated with the same field. In the second case, we regard the p-delimiter as being simply a character that appears in values of a field, and thus, the two blocks it separates are likely to be associated with the same field. 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. Then, we can analyze how common a p-delimiter is for each field and how they typically behave; that is, which of the cases (a or b) apply.

For instance, in Figure 2, 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" with *Title* rather than with *Author*. These ideas are elaborated in the following sequence.

Consider the sequence $l, p_0, u_1, p_1, \ldots, u_n, p_n, r$, where l and r are matched blocks associated with 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 with m_l or to m_r. First, we consider that only one of the p-delimiters 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 $(0 < j \leq i)$ with m_l (i.e., the same field as the block on the left), and all u_k $(i > k \geq n)$ with m_r (i.e., the same field as the block on the right).

Now, consider the following expressions:

$$T(p_k, m_l, m_r) = \frac{f(p_k, m_l, m_r)}{\sum_{p_j \in P} f(p_j, m_l, m_r)} \tag{4}$$

where $f(p, m_l, m_r)$ is the frequency of p-delimiter p between contiguous blocks associated with fields m_l and m_r by the matching step, and P is the set of all p-delimiters.

$$C(p_k, m) = \frac{f(p_k, m)}{\sum_{p_j \in P} f(p_j, m)} \tag{5}$$

where $f(p, m)$ is the frequency of a p-delimiter p between contiguous blocks associated with the same field m by the matching step, and P is the set of all p-delimiters.

Intuitively, Equation 4 estimates the probability of a given p-delimiter p_i to be a delimiter between fields m_l and m_r while Equation 5 estimates the probability of p_i to be a character occurring as part of the values of a field m. Note that the frequencies used in these equations are obtained after analyzing each p-delimiter in all citations to be extracted. This is done to ensure that meaningful statistics on the role and position of the p-delimiter are produced.

In our method, these factors are considered for deciding which p-delimiter p_i is the field delimiter in the sequence. For this, we use Equation 6, defined as follows.

$$\begin{aligned} D(p_k, m_l, m_r) = 1 - \Bigg[&(1 - T(p_k, m_l, m_r) \\ &\times \prod_{0 \leq j < k} 1 - C(p_j, m_l) \\ &\times \prod_{k > j \geq n} 1 - C(p_j, m_r) \Bigg] \end{aligned} \tag{6}$$

where p_k is a p-delimiter and k is its ordinal position.

Given a delimiter p_k, Equation 6 takes into account (a) the probability of p_k to be a typical field delimiter between values of m_l and m_r, (b) the probability of the p-delimiters on the left of p_k to be part of the values of field m_l, and (c) the probability of the p-delimiters on the right of p_k to be part of the values of field m_r.

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 field delimiter is selected as the one for which this equation gives the highest value.

In Figure 2e, for instance, the block containing the term "Bossa Nova" is associated with *Title* since D(":", *Title*, *Author*) < D(".", *Title, Author*).

Joining

When the binding step is over, each block in the citation string is associated with a metadata field. Then, the last step in our extraction method consists of joining together blocks associated with a same field to form the values of that field. For most of the cases, this step is straightforward to accomplish since it simply requires joining contiguous blocks associated with a same field. However, joining blocks associated with the *Author* field requires a more careful procedure since there may be several *Author* values on a citation string. Thus, in this section, we describe how we handled joining blocks to form values for the *Author* field. For instance, the *Author* blocks in Figure 2e, must be joined to form *Author* values illustrated in Figure 2f.

The solution for this problem relies on the information available on the knowledge base. Let η be the average number of terms in the occurrences of the *Author* field in the knowledge base. We assume that the number of terms found in the values of *Author* in any citation string is approximately equal to η.

Now, consider that there is some set s of strings used as implicit delimiters for separating the values of *Author* in a citation string. For instance, in a given citation, the string "," may be used as a delimiter for all values of *Authors*, except for the last value which is separated by the string "and". In this case, $s = \{$",", "and"$\}$. In our method, as mentioned earlier, we rely on the observation that the number of terms bounded by the strings in s should be approximately equal to η.

Consider a sequence of blocks that must be joined to compose the values of *Author*. Given a set of delimiter strings s, two or more contiguous blocks should be joined if the p-delimiter p between them is not a delimiter string (i.e, $p \notin s$). Hence, we must determine which p-delimiters compose s.

The solution we adopt is to take candidate sets of delimiters and, for each candidate set, evaluate if this set is the one that results in values of *Author* with a number of terms closest to η. For this, we define a metric we call *delimiting error*, which is based on the difference between the lengths of the values (in number of terms) and the average length found in the knowledge base (η).

$$de(s, a, \eta) = \prod_{x \in split(s,a)} dif(len(x), \eta) \tag{7}$$

where s is a set of delimiters, a is the portion of the citation string composed by *Author* blocks, and the following auxiliary functions are used:

- $split(s, a)$ returns all substrings of a that are bounded by some delimiter $p \in s$.

- $len(x)$ returns the number of terms in the string x.
- $dif(l_1, l_2) = |l_1 - l_2|$ if $l_1 \neq l_2$, and $dif(l_1, l_2) = \varepsilon_0$ otherwise, where ε_0 is a small constant.

Intuitively, given a citation string with a set of *Author* blocks to be joined, Equation 7 calculates a score based on the distance between η and the number of terms of each *Author* value obtained when using s as the set of delimiters.

Thus, let P be the set of p-delimiters between *Author* blocks. We evaluate the delimiting error for each subset of p-delimiters $s \subseteq P$ using Equation 7. The set of delimiters used for *Author* values will be the one with the smallest delimiting error.

As an example, consider the citation in Figure 2e, in which the set of delimiters between *Author* blocks is ".", ",". Also assume $\eta = 2.7$.[2] When the delimiter "," is used as a separator of *Author* values, the delimiting error is about 0.21. In contrast, using the delimiter "." or the delimiter set ".", ",", the delimiting error is about 0.83 in both cases. Thus, the delimiter "," is the best choice. In Figure 2f, we show the *Author* values obtained with this delimiter.

Experiments

In this section, we present the experiments performed to evaluate our approach on the task of extracting metadata from bibliographic citation strings. We also present an experimental comparison between our proposed method, FLUX-CiM, and CRF (McCallum, 2006), a method regarded as the state-of-the-art in bibliographic data extraction from research papers.

In all experiments, we carried out similar extraction tasks over citations from three distinct domains: Health Sciences (HS), Social Sciences (SS), and Computer Science (CORA). In all cases, we use samples of citation records of each specific domain to generate the knowledge bases. Then, we executed extraction processes over a set of citations strings from the same domain. Table 1 presents some features of each collection that we used in our experiments. Note that the number of metadata fields in CORA varies from 1 to 13. This happens because the citation strings in this collection come from different sources such as conference papers and journal papers from several distinct publishers, and thus, they have distinct citation styles.

Setup

CORA is a heterogeneous collection composed by 500 assorted citations from several computer science conferences and journals, and was previously used by McCallum (2006) to evaluate CRF. We randomly choose 350 citations to generate the knowledge base and another 150 different citations to test our method. This proportion was the same as that used in McCallum (2006) to evaluate CRF.

TABLE 1. Features of the collections used in the experiments.

Domain	Knowledge base size	No. of fields	No. of citations
HS	5,000	6	2,000
SS	5,000	6	2,000
CORA	350	1–13	150

HS = Health Sciences; SS = Social Sciences; CORA = Computer Science.

For experiments in the HS domain, we used a collection of citations from PMC. For the SS domain, the collection was obtained from the Scielo Digital Library (http://www.scielo.org/). For each of these domains, we used collections composed of more than 50,000 citation records. The HS and the SS collections are both considered as *well-organized* since their citation strings follow a uniform style and as *controlled* since for each citation string there is a structured metadata record where the fields of the citation string are explicitly identified. Therefore, by carrying out experiments over these controlled collections, we can automatically verify the extraction results for a large number of citation strings. The knowledge bases were randomly built with 5,000 citations while the extraction process used 2,000 different citations. In so doing, we ensured that there was no overlap between the knowledge base and the citations set.

The HS and the SS collections also were used to perform an experiment that evaluates how our method behaves when the number of citation records in the knowledge base varies. For this experiment, we vary the size of the knowledge base from 50 to 10,000 citation records. Note that we were unable to perform this experiment with CORA due to the small number of citations available.

All experiments we report in this section were repeated five times. Thus, each value presented here represents the average of the values obtained in each of the five repetitions. In our experiments, we evaluated the extraction results obtained after the whole extraction process.

In the evaluation, we used the well-known precision, recall, and F-measure metrics, which are computed as follows. Let B_i be a reference set and S_i be a test set to be compared with B_i. We define precision (P_i), recall (R_i) and F-measure (F_i) as:

$$P_i = \frac{|B_i \cap S_i|}{|S_i|} \quad R_i = \frac{|B_i \cap S_i|}{|B_i|} \quad F_i = \frac{2(R_i.P_i)}{(R_i + P_i)} \tag{8}$$

Results

Verifying the Blocking Hypothesis

The first result we report aims at verifying in practice the hypothesis we have formulated regarding blocking: In general, in a citation string, every field value is bound by a p-delimiter, but not all p-delimiters bound a value. To verify this, we look into the citation in the collections we used for the experiments and count the field values that are bound by some p-delimiter. As expected, in all datasets, 100% of the field values were bound by a p-delimiter.

[2] This is the actual value we found in one of the citation collections used in our experiments.

TABLE 2. Block-level precision and recall for each field after both the matching and the binding steps for the Computer Science (CORA), Health Sciences (b), and Social Sciences (c) domains. The percentage of unmatched blocks after the matching step also is presented.

	Matching				Binding		
Field	Precision (%)	Recall (%)	*F*	Unmatched blocks (%)	Precision (%)	Recall (%)	*F*
(a) CORA							
Author	99.78	79.29	0.8836	20.63	99.82	98.96	0.9939
Title	98.11	90.43	0.9412	7.83	97.19	97.61	0.9740
Journal	95.80	97.86	0.9682	1.43	95.80	97.86	0.9682
Date	99.70	97.38	0.9853	2.04	97.98	99.13	0.9855
Pages	97.87	98.71	0.9829	1.29	97.06	99.14	0.9809
Conference	100.00	96.00	0.9796	0.40	99.18	96.40	0.9777
Place	98.88	89.85	0.9415	9.64	98.48	98.48	0.9848
Publisher	100.00	100.00	1.0000	0.00	100.00	100.00	1.0000
Number	97.87	97.87	0.9787	2.13	97.87	97.87	0.9787
Volume	100.00	98.25	0.9912	0.00	100.00	98.25	0.9912
Average	98.80	94.56	0.9652	4.54	98.34	98.3	0.9835
(b) Health Sciences							
Author	99.04	94.33	0.9663	4.96	98.89	99.26	0.9907
Title	93.71	90.54	0.9210	6.17	92.90	95.96	0.9441
Journal	97.51	89.22	0.9318	2.22	97.15	89.32	0.9307
Date	99.85	96.89	0.9835	0.00	99.85	96.89	0.9835
Pages	99.90	98.54	0.9922	0.00	99.80	98.54	0.9917
Volume	98.53	97.65	0.9809	0.00	98.53	97.65	0.9809
Average	98.09	94.53	0.9626	2.22	97.86	96.27	0.9703
(c) Social Sciences							
Author	99.35	95.26	0.9726	3.56	99.01	99.87	0.9044
Title	92.14	94.78	0.9344	5.89	91.17	98.43	0.9466
Journal	98.22	94.41	0.9628	2.05	97.05	94.99	0.9601
Date	99.57	97.01	0.9827	0.00	99.57	99.01	0.9827
Pages	99.65	98.45	0.9905	0.00	99.65	98.45	0.9905
Volume	98.67	98.66	0.9866	0.00	98.67	98.66	0.9866
Average	97.93	96.43	0.9716	1.91	97.52	97.90	0.9768

Block-Level Results

We now present results that show how correctly the blocks were associated with their respective fields in our method.

Consider the set of citation strings we used for evaluating the extracting process in a given domain. Let B_i be the set of all blocks in the strings in this set which compose the values of a metadata field m_i. These blocks were used as references to our block-level verification.

Now, let S_i be the set of blocks associated with m_i after a given step of our method (e.g., matching or binding). The precision and recall obtained with Equation 8 for these experiments are presented in Table 2 for the CORA(a), the HS (b), and the SS (c) domains. To compare the outcome of the first two steps of our method, we separately present the results obtained after the matching step and after the binding step, which are cumulative. We also present the number of blocks which were left unmatched after the matching step.

In the Methods, we argued that the matching step is the main step of our approach. To verify this, note that on average, less than 5% of the blocks are left unmatched for all sets of citations. This occurs because any block that presents at least one of its terms occurring on the knowledge base are matched. However, this fact alone would be not enough to guarantee the high precision and recall results obtained, which are due to the suitability of the FF function (Equation 1) we have proposed for the matching.

The results in Tables 2a to 2c also show that the binding step plays an important role in our method since it was able to significantly improve the results of recall by keeping precision levels very similar to the ones in the matching step.

Overall, there was a single case in which the matching step was not able to distinguish with very high accuracy the blocks from two distinct fields. This occurred for the fields *Title* and *Journal* in the HS domain. This can be explained by the large number of common terms between these two fields.

We should stress the high-quality levels achieved in CORA even though this collection contains citations in various styles and the fact that the knowledge base was relatively small in size.

Field-Level Results

To demonstrate the effectiveness of the whole extraction process with our method, we evaluate the extraction quality after the joining step, in which blocks are joined to compose the values of fields. Here, instead of blocks, we analyze for each field occurring in the citations if the values assigned by our method to this field are correct. This is

TABLE 3. Field-level precision and recall for each field after the joining step for the CORA (a), HS (b) and SS (c) domain.

Field	Precision (%)	Recall (%)	F-measure
(a) CORA			
Author	97.21	98.67	0.9793
Title	93.01	96.67	0.9480
Journal	93.45	91.5	0.9246
Date	96.01	90.01	0.9291
Pages	97.98	98.81	0.9839
Volume	100.00	99.14	0.9957
Tech	94.67	91.9	0.9326
Location	90.01	94.78	0.9233
Note	91.00	88.81	0.8989
Book title	93.16	91.73	0.9244
Editor	90.76	91.66	0.9121
Publisher	90.00	96.56	0.9316
Institution	92.15	91.11	0.9163
Average	96.28	95.8	0.9601
(b) HS Domain			
Author	98.57	99.04	0.9880
Title	84.88	85.14	0.8501
Journal	97.23	89.35	0.9312
Date	99.85	99.50	0.9967
Pages	99.70	99.20	0.9945
Volume	96.41	98.75	0.9757
Average	96.11	95.16	0.9560
(c) SS Domain			
Author	96.48	99.17	0.9781
Title	91.20	96.67	0.9386
Journal	97.99	93.68	0.9579
Date	99.57	97.01	0.9827
Pages	99.65	98.45	0.9905
Volume	98.67	98.66	0.9866
Average	97.26	97.27	0.9724

important especially for the *Author* field to check if the blocks associated with this field were correctly joined (i.e., if terms from the same author names were joined in the same field value).

In this case, we redefine Equation 8 by considering B_i the set of complete values of m_i and S_i the set of complete values associated with m_i by our method. Again, the B_i sets were automatically obtained for all domains. The results are shown in Table 3 for the CORA (a), HS (b), and SS (c) domains. Note that precision and recall are defined here for complete field values. Thus, if at least one block of the m_i value was not associated with m_i, we consider that all the m_i value was incorrectly extracted.

From Table 3a to 3c, note that the high accuracy levels reached after the matching and binding steps remain after the joining steps. The exception was the F-measure value for the field *Title* from the HS, which was around 0.85. A closer look at the values of this field revealed a large overlap with the terms in the values of the *Journal* field in this domain. Because of this overlap, some *Journal* blocks were wrongly associated with *Title* in the matching step. This can be observed by looking at the recall value for *Journal* (89.32%) and the precision value for *Title* (93.7%) after the matching step in Table 2b, which are relatively low. This situation was propagated through the binding step until the joining step.

TABLE 4. Average citation-level precision and recall for citations after the joining step.

Domain	Precision (%)	Recall (%)	F-measure
Health Sciences	94.82	95.10	0.9496
Social Sciences	97.32	97.21	0.9726
CORA	92.14	94.78	0.9344

Citation-Level Results

The final aspect we analyzed in our experiments is how well each citation record was extracted by our method; that is, we want to verify whether the fields composing each record were correctly extracted. Note that while the field-level results presented earlier involve all values from a given field regardless of the citations in which they occur, we examine in this section the extraction results on a per-citation basis, averaging the results.

To present these results, consider each reference set B_i as the set of field values in a given citation record C_i. Now, let S_i be the set of field values extracted for C_i by our method. Then, precision and recall are calculated using Equation 8. In Table 4, we present the average of precision and recall obtained in the experiments for all domains.

The values in Table 4 were obtained by taking into consideration all field values occurring in each citation, which may vary for each individual citation. These results demonstrate that our method is able to deal with a variety of citation types, without having to rely on a predefined set of citation styles.

In our final experiment in this section, we verify how our method behaves when the size of the knowledge base varies. The results of this experiment are presented in Figure 3, in which for the HS and SS domains we used an increasing number of sample citation metadata records (from 50 to 10,000) and calculated the citation-level F-measure resulting from running the extraction process over each collection. Note that in both cases, F-measure values quickly stabilize, reaching over 0.95 with 3,000 sample citation records in the knowledge base, and this value remains the same until 10,000 sample citation records. This shows that our method does not require a large knowledge base to reach a good extraction quality in the HS and SS collections we used. Again, we were unable to perform this experiment with CORA due to the small number of citations available in this collection.

Discussion

Although the experimental results we have presented here demonstrate the high effectiveness of our proposed method, the problem of citation extraction is still a challenge, mainly due to some pathological cases that would prevent any method from achieving a perfect result.

In Figure 4, we present two examples of real bibliographic citations and their respective extraction results produced by

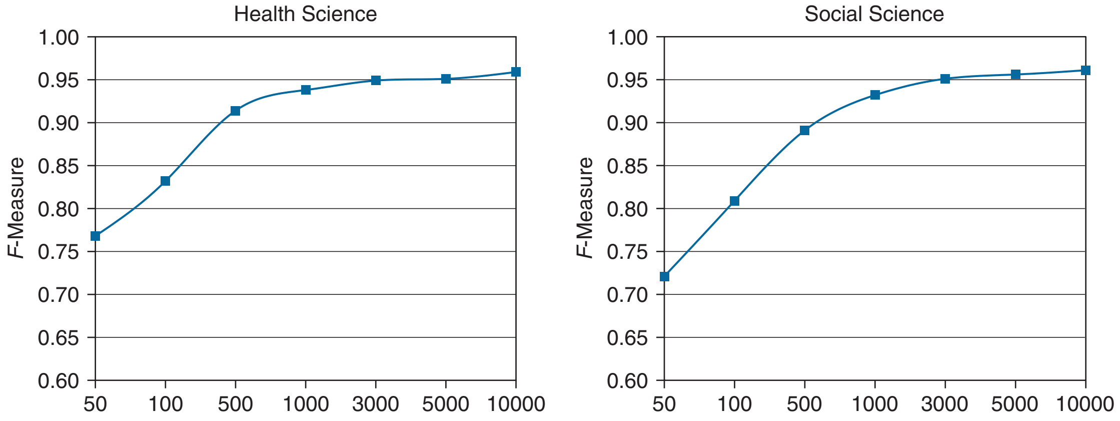


FIG. 3. Performance of the citation extraction relative to the size of the knowledge base for the Health Sciences (HS) and Social Sciences (SS) domains.

FLUX-CiM. In the first citation, Figure 4a, one of the values of the field *Author*, "Pathology Review Committee," was misidentified in the extraction process as a value of field *Title* since the term "Pathology" is typical of this field.

In the bibliographic citation presented in Figure 4b, it is even hard for humans to correctly separate the author names. We searched for other citations of the same paper and found that the correct author values are "Clayton Lewis," "D. Charles Hair," and "Victor Schoenberg." Note that the first value was represented distinctly from the other two.

Comparing FLUX-CiM and CRF

Experimental Comparison

We now present the results of an experimental comparison we conducted between FLUX-CiM and CRF, a state-of-the-art method for citation extraction. Note that the output provided by CRF is slightly distinct from that provided by FLUX-CiM, in the sense that values in multiple-valued fields (e.g., author names) are not individually separated. Thus, to ensure a fair comparison between the two methods, the results we report here are expressed according to the same metrics used in McCallum (2006). For this, we redefine Equation 8 by considering B_i as the set of correct values of a field m_i and S_i as the set of values associated with m_i by the extraction method. Distinct from the field-level results presented earlier, in this case, each of the multivalued fields (i.e., *Authors*) are considered as a single-valued field (see Table 5).

For all three collections, we ran an implementation of publicly available CRF (http://crf.sourceforge.net), which was implemented according to McCallumImplement. To ensure a fair comparison, the same set of citation records was used to train the CRF model and to generate the knowledge base in FLUX-CiM. Similarly, the same citation strings in the test set were applied for both methods.

For the HS and SS collections, each result presented was gathered after five complete executions; that is, in each execution, a training set for CRF, a knowledge base for FLUX-CiM, and a test set for both methods were randomly generated. For CORA, as in McCallum (2006), experiments were executed only once since the number of citations available in this collection is too small to allow nonoverlapping executions.

TABLE 5. Comparative F-Measure results for CORA (a), Health Sciences (b), and Social Sciences (c).

Field	FLUX-CiM	CRF	t test (%)	Wilcoxon (%)
(a) CORA				
Author	0.9420	0.9940	–	–
Title	0.9357	**0.9830**	2.00	2.00
Journal	**0.9262**	0.9130	1.00	1.00
Date	0.9566	**0.9890**	3.00	5.00
Pages	0.9567	0.9860	–	–
Book title	0.9364	0.9370	–	–
Location	**0.9315**	0.8720	1.00	1.00
Publisher	**0.9250**	0.7610	1.00	1.00
Others	**0.9408**	0.8940	1.00	1.00
Average	**0.9390**	0.9254	3.00	1.00
(b) Health Sciences				
Author	**0.9662**	0.9548	4.00	2.00
Title	**0.9956**	0.9616	1.00	1.00
Journal	**0.9371**	0.8930	1.00	1.00
Date	**0.9987**	0.9657	2.00	2.00
Pages	0.9783	0.9647	–	–
Volume	**0.9995**	0.9592	1.00	1.00
Average	**0.9792**	0.9498	1.00	1.00
(c) Social Sciences				
Author	**0.9954**	0.9431	1.00	1.00
Title	**0.9978**	0.9714	1.00	1.00
Journal	**0.9401**	0.8889	1.00	1.00
Date	**0.9984**	0.9619	3.00	5.00
Pages	**0.9318**	0.9067	1.00	1.00
Volume	**0.9720**	0.9214	1.00	1.00
Average	**0.9726**	0.9322	1.00	1.00

CRF = Conditional Random Fields.

For the HS and SS collections, we used 5,000 citations for the knowledge base and for training the CRF, and 2,000 citations for the testing. For CORA, we used a knowledge base with 350 citation records, the same number for training the CRF, and then 150 citations to test both methods.

For all comparisons reported, we used the Wilcoxon signed-rank test (Wilcoxon, 1945) and the Student's t test (Anderson & Finn, 1996) for determining if the difference

	Real Bibliographic Citation	Extraction Result
(a)	Nagtegaal ID, Klein Kranenbarg E, Hermans J, van de Velde CJH, van Krieken JHJM, Pathology Review Committee. Pathology data in the central database of multicenter randomized trials need to be based on pathology reports and controlled by trained quality managers. J Clin Oncol. 2000;18:1771-1779.	***Author 1:*** Nagtegaal ID ***Author 2:*** Klein Kranenbarg E ***Author 3:*** Hermans J ***Author 4:*** van de Velde CJH ***Author 5:*** van Krieken JHJM ***Title:*** Pathology Review Committee. Pathology data in the central database of multicenter randomized trials need to be based on pathology reports and controlled by trained quality managers. ***Journal:*** J Clin Oncol. ***Year:*** 2000 ***Volume:*** 18 ***Pages:*** 1771–1779
(b)	Lewis, Clayton, D. Charles Hair, Victor Schoenberg (1989). Generalization Consistency Control. In Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems. pages 1-5.	***Author 1:*** Lewis, Clayton, D ***Author 2:*** Charles Hair ***Author 3:*** Victor Schoenberg ***Year:*** 1989 ***Title:*** Generalization Consistency Control ***Conference:*** In Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems ***Pages:*** pages 1–5

FIG. 4. Examples of pathological cases in bibliographic citations from the Health Sciences (HS) (a) and Computer Science (CORA) (b) domains and their respective extraction results.

in performance was statistically significant. In all cases, we only drew conclusions from results that were significant at least at a 5% level for both tests. Nonsignificant values are omitted.

By observing the results presented in Table 5 note that in all three distinct datasets, our method achieved higher results than the CRF in most of the fields, according to both statistical tests. The better results achieved by the CRF in two fields of CORA (boldfaced) can be explained by the limited number of citation records in the knowledge base for this collection. In the HS and SS collections, where larger sets of citation records are available to test and generate the knowledge base, FLUX-CiM performed better than did CRF for all fields. These experiments demonstrate that even without any user intervention to create a training set, FLUX-CiM achieves better extraction quality than do CRFs.

TABLE 6. Examples of the citations styles and final configuration of each set.

	Citation style example
1	Kerlikowske K, Orel SG, Troupin RH. Nonmammographic imaging. Semin Roentgenol. 1993;28:231–241
2	231–241: Nonmammographic imaging. Kerlikowske K: Orel SG: Troupin RH, 1993; 28. Semin Roentgenol
3	1993; Kerlikowske K; Orel SG; Troupin RH; Semin Roentgenol. Nonmammographic imaging. 231–241: 28
4	Nonmammographic imaging: 1993, Kerlikowske K, 231–241, Orel SG; Troupin RH. Semin Roentgeno

Set	No. of citations per style	No. of citations in the set
1 style	2,000	2,000
2 styles	1,000	2,000
3 styles	667	2,001
4 styles	500	2,000

Dealing With Different Citation Styles

As already discussed, one of the features we regard as very important in FLUX-CiM is its flexibility in extracting from citations regardless of the particular style used. This happens because our extraction approach does not rely on patterns encoding specific delimiters used in a particular citation style but rather on features of the citation fields and their values.

To evaluate such a property, we performed a set of experiments in which the test sets include citation strings with distinct styles. These experiments simulate situations in which citations are to be extracted from several scientific papers from different sources with different citation styles.

In the experiments, test sets were built as follows. We take citation strings from the HS and SS collections and generate four test sets, such that set i contains $\lceil N/i \rceil$ citations formatted according to style i, where $N = 2{,}000$ and $1 \geq i \leq 4$. Style 1 corresponds to the original citation style used in each collection. The other styles were generated by randomly changing the implicit field delimiters and the relative order of the fields. By randomly generating citation styles, we aimed to simulate situations in which a previously unseen citation style is deployed.

In Table 6, we present examples of the styles used (first panel) and a summary of the final configuration of each test set (second panel). For building a knowledge base for FLUX-CiM and the training CRF model, we randomly took 5,000 citation records in their original citation style (i.e., Style 1) from each respective collection.

The results of this experiment are presented in Table 7. Note that the F-measure obtained with CRF decreases with an increase in the number of citations styles. This happens because the CRF model looks for specific features learned from a single style that it was trained on. On the other hand, with FLUX-CiM, the F-measure remains constant regardless of the number of different citation style used, thus corroborating our claims about the flexibility of our method.

TABLE 7. F-measure values achieved with different citation styles for the Health Sciences (a) and Social Sciences (b) collections.

No. of styles	FLUX-CiM	CRF	Wilcoxon (%)	t (%)
(a) Health Sciences				
1	0.9792	0.9498	1.00	1.00
2	0.9792	0.7065	1.00	1.00
3	0.9792	0.4033	1.00	1.00
4	0.9792	0.3567	1.00	1.00
(b) Social Sciences				
1	0.9704	0.9322	1.00	1.00
2	0.9704	0.7586	1.00	1.00
3	0.9704	0.3867	1.00	1.00
4	0.9704	0.3199	1.00	1.00

Extraction Feedback

The experimental results we have presented thus far demonstrate the high extraction quality and high levels of flexibility achieved by FLUX-CiM; however, for this to occur, it is very important that the underlying knowledge base covers a representative portion of the domain of interest. Indeed, as discussed and demonstrated in this article, the size of the knowledge base directly influences the quality of the extraction. On the other hand, there could be cases in which new features must be incorporated to the knowledge base from time to time to reflect a new trend found on the target domain. For instance, the term "Bluetooth" was only recently incorporated in the vocabulary of the computer science domain. Such a phenomenon also can occur for values of fields such as venues and authors.

To cope with both requirements (i.e., having a significant number of citations in the knowledge base and guaranteeing the representativeness of these citations with respect to the current state of the target domain), it would be necessary to collect data from this domain (e.g., on the Internet) and add this data to the knowledge base, as described earlier. Although very simple, such a task would still require the user's intervention, which could become inconvenient in a scenario in which autonomy is required.

In this section, we propose a solution to this problem by directly incorporating the results of the extraction processes to the knowledge base, a process we call *Extraction Feedback*, which is illustrated in Figure 5. Consider a knowledge-base K on a given domain D. Now, suppose we use FLUX-CiM to extract a certain amount of citations from a set of sources S, also on domain D. The Extraction Feedback consists of taking the terms composing the data values extracted from S for each field and updating the corresponding fields in K.

At a first sight, using the data extracted to directly update the knowledge base could introduce a certain amount of noise to it that could compromise the results of extraction processes based on the updated knowledge base. However, as shown earlier, FLUX-CiM achieves a good extraction quality even with a small knowledge base. Thus, we argue that use of the extraction outcome to perform the Feedback process can be quite "safe" since the possible amount of noise is very low.

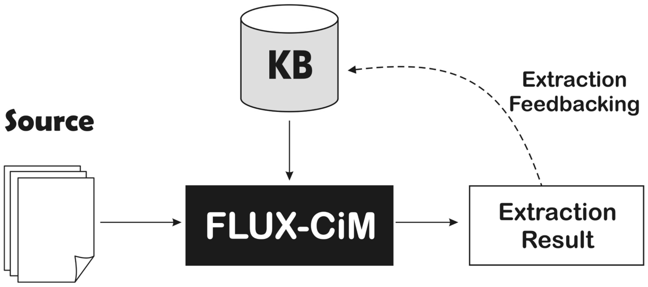


FIG. 5. Extraction Feedback.

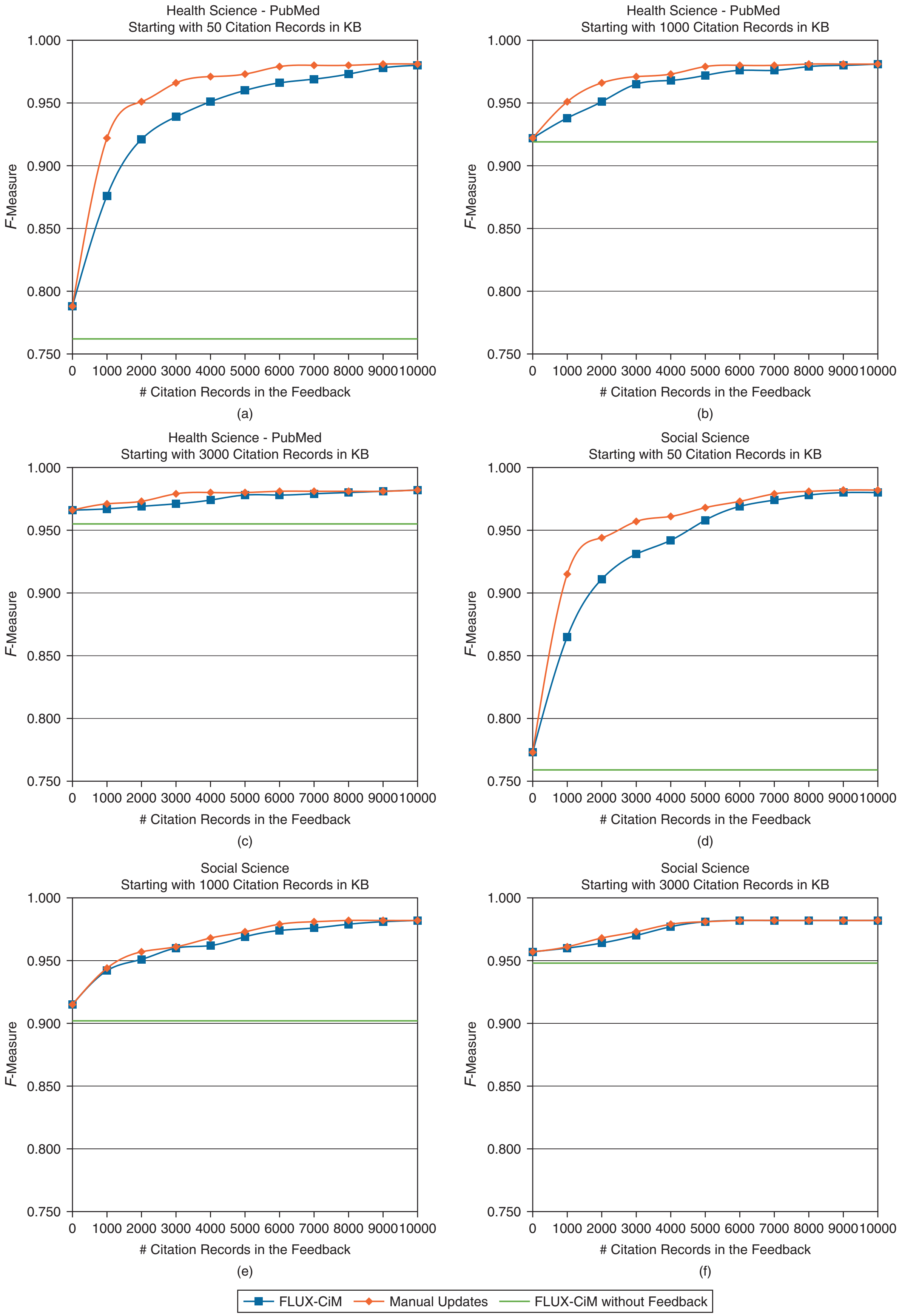
To corroborate this claim, we present the results of experiments performed with the Extraction Feedback process.

In the experiments reported next, we aimed at showing the effectiveness of the Extraction Feedback process to automatically update and expand the knowledge base without compromising the extraction quality. We have conducted similar experiments in the HS and SS domains since these are the domains for which we have a large set of available citations. We analyzed the behavior of our method using Extraction Feedback in three different scenarios. First, we started with a knowledge base constructed using only 50 citation records, and then executed the extraction task into a source containing 1,000 citation strings. For each run, we evaluated the extraction quality in terms of F-measure. We also performed experiments starting with knowledge bases built with 1,000 and 3,000 citation records. In all cases, citation records for the initial bases and for the automatic Feedback were randomly selected. Additionally, we ran each experiment five times. The results of the experiments are presented in Figure 6, in which each point represents the average of the values obtained in each of the five runs.

In Figure 6, each graph shows the quality achieved by FLUX-CiM in terms of F-measure as a function of the number of citation records used for updating the knowledge base according to the automatic Extraction Feedback process.

This quality is compared with the quality level that would be achieved if the knowledge base were manually updated with same number of totally correct citations records. For this, we took the corresponding correct citation records and added them to the knowledge base. This represents an *upper bound* for the quality that can be achieved after using the Feedback. The straight line in these graphs represents the extraction quality if the extraction process were executed in the whole test set; that is, if the FLUX-CiM had been used to extract 10,000 citation strings with only the initial knowledge base.

Note that in all distinct scenarios, the automatic Extraction Feedback process with 9,000 citation records or more reaches the upper bound quality levels. This means that even when starting with a small set of citation records in the knowledge base, it is possible to use the Extraction Feedback process in a automatic fashion without user intervention to reach high-quality results. Furthermore, even when starting with a small knowledge base, it is better to perform the extraction task in small test sets, then in the whole citation strings set available.



Health Science - PubMed Starting with 50 Citation Records in KB

(a)

Health Science - PubMed Starting with 1000 Citation Records in KB

(b)

Health Science - PubMed Starting with 3000 Citation Records in KB

(c)

Social Science Starting with 50 Citation Records in KB

(d)

Social Science Starting with 1000 Citation Records in KB

(e)

Social Science Starting with 3000 Citation Records in KB

(f)

FIG. 6. Behavior of the citation extraction performance with the Extraction Feedback the Health Sciences (HS) and Social Sciences (SS) domains.

In the graphs of Figure 6c to f, in which the initial knowledge bases were built using 1,000 or 3,000 citations records, the automatic Feedback process brings the same quality improvement that would be obtained with manual perfect updates to the knowledge base.

These results corroborate our claim that even if our Extraction Feedback may introduce some errors in the knowledge base since the extraction quality obtained by FLUX-CiM is not perfect, it is good enough to not compromise extraction processes carried out after the Feedback. This indicates that updating the knowledge base can be accomplished automatically, with no user intervention.

Conclusions and Future Work

In this article, we have proposed a novel method, FLUX-CiM, for extracting components (e.g., author names, article titles, venues, page numbers) from bibliographic citations. Unlike previous methods in the literature, our method does not rely on patterns encoding specific delimiters used in a particular citation style. This feature yields a high degree of automation and flexibility, and allows FLUX-CiM to extract from bibliographic citations in any given citation style, as demonstrated by experiments in this article. FLUX-CiM relies on a knowledge base automatically constructed from an existing set of sample metadata records of a given domain (e.g., CORA, HS, SS, etc.). These records usually are easily available on the Web or other public data repositories.

Our method differs from related knowledge-based approaches that rely on manually built knowledge bases for recognizing the components of a citation. In addition, FLUX-CiM works differently from previous methods based on models learned from user-driven training. The extraction process in our method is based on (a) estimating the probability of given term found on a citation string to occur as a value of a given citation field according to the information encoded in a knowledge base, and (b) the use of generic structural properties of bibliographic citations.

The effectiveness and applicability of our proposed method were demonstrated by experiments for extracting information from bibliographic citations in scientific papers of three distinct domains: HS, CORA, and SS. The experiments in this article show that FLUX-CiM obtains precision and recall levels over 95% for the fields present in the set of citations, and an average recall of over 94% for the fields present in each citation.

We also performed an experimental comparison between FLUX-CiM and CRF. The results of these experiments demonstrated that even without any user intervention to create a training set, FLUX-CiM achieves better extraction quality than does CRF.

The flexibility of FLUX-CiM was experimentally verified by means of a set of experiments in which the test sets include citation strings with different styles. The results of these experiments corroborate our claim that the extraction quality remains steady regardless of the number of different citation styles used.

Finally, we proposed a process, Extraction Feedback, for automatically updating and expanding the knowledge base by directly incorporating the results of an extraction process on it carried out by FLUX-CiM. We have shown that such a strategy can be used to achieve good extraction results, even if only a very small initial sample of bibliographic records is available for building the knowledge base. Despite the introduction of some errors in the knowledge base by this process, the quality of the results obtained shows that this does not compromise extraction processes carried out after the Feedback. In effect, we have demonstrated that with FLUX-CiM, updating the knowledge base can be accomplished automatically, with no user intervention.

For future work, we intend to investigate different matching functions that might better distinguish citation fields that have common values to describe their domains (e.g., author's name and editor's name). This kind of function could make our method more general and robust.

An interesting strategy to achieve higher extraction results in more complex types of data would be the use of our method to automatically discover the implicit style of the data and then use this property as evidence in, for example, another supervised extraction process.

We also may investigate the applicability of our method for extracting citations form sources other than citation lists from papers. For instance, it seems interesting to have a mechanism to automatically populate a digital library with metadata directly from Web sites of recent conferences or from the headers (title, authors, abstract) of the papers published in these venues.

Acknowledgments

This work was partially supported by Grant 551013/2005-2 CNPq from Projects 5S-VQ, Grant 550874/2007-0 CNPq from InfoWeb, Grant 573871/2008-6 CNPq from INCTWeb, Grant 55.3126/2005-9 CNPq from SIRIAA, and Grant 575553/2008-1 CNPq from MinGroup; by individual CNPq fellowship grants to Edleno S. de Moura, Altigran S. Silva, and Marcos André Gonçalves; by a FAPEAM scholarship to Eli Cortez; and by a "UOL Bolsa Pesquisa" program grant (Process 20080131152900).

References

Agrawal, S., Chaudhuri, S., Das, G., & Gionis, A. (2003). Automated ranking of database query results. Proceedings of the CIDR 2003 Biennial Conference on Innovative Data Systems Research.

Anderson, T., & Finn, J. (1996). The new statistical analysis of data. (1st ed.). New York: Springer-Verlag.

Arasu, A., & Garcia-Molina, H. (2003). Extracting structured data from web pages. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (pp. 337–348). New York: ACM Press.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30(1–7), 107–117.

Calado, P., Cristo, M., Gonçalves, M.A., de Moura, E.S., Ribeiro-Neto, B., & Ziviani, N. (2006). Link-based similarity measures for the classification of

DOI: 10.1002/asi

web documents. Journal of the American Society for Information Science and Technology, 57(2), 208–221.

Cortez, E., da Silva, A., Gonçalves, M., Mesquita, F., & de Moura, E. (2007). FLUX-CIM: Flexible unsupervised extraction of citation metadata. Proceedings of the 2007 Conference on Digital libraries (pp. 215–224).

Couto, T., Cristo, M., Gonçalves, M.A., Calado, P., Ziviani, N., Moura, E., & Ribeiro-Neto, B. (2006). A comparative study of citations and links in document classification. In Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 75–84). New York: ACM Press.

Crescenzi, V., Mecca, G., & Merialdo, P. (2001). Roadrunner: Towards automatic data extraction from large web sites. In Proceedings of the 27th International Conference on Very Large Data Bases (pp. 109–118). San Francisco: Kaufmann.

Culotta, A., Kristjansson, T.T., McCallum, A., & Viola, P.A. (2006). Corrective feedback and persistent learning for information extraction. Artificial Intelligence, 170(14–15), 1101–1122.

Day, M.-Y., Tsai, T.-H., Sung, C.-L., Lee, C.-W., Wu, S.-H., Ong, C.-S., & Hsu, W.-L. (2005). A knowledge-based approach to citation extraction. In Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (pp. 50–55). New York: IEEE Systems, Man, and Cybernetics Society.

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

Freitag, D., & McCallum, A. (2000). Information extraction with HMM structures learned by stochastic optimization. In Proceedings of the 17th National Conference on Artificial Intelligence (pp. 584–589). Menlo Park, CA: AAAI.

Gonçalves, M.A., Moreira, B.L., Fox, E.A., & Watson, L.T. (2007). What is a good digital library? Defining a quality model for digital libraries. To appear in Information & Process Management, 43(5), 1416–1437.

Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., & Fox, E.A. (2003). Automatic document metadata extraction using support vector machines. In Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 37–48). Washington, DC: IEEE Computer Society.

Hsu, C.-N., & Dung, M.-T. (1998). Generating finite-state transducers for semi-structured data extraction from the web. Information Systems, 23(9), 521–538.

Hu, Y., Li, H., Cao, Y., Meyerzon, D., & Zheng, Q. (2005). Automatic extraction of titles from general documents using machine learning. In Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, Tools & Techniques: Supporting Classification (pp. 145–154).

Kushmerick, N. (2000). Wrapper induction: Efficiency and expressiveness. Artificial Intelligence, 118(1–2), 15–68.

Laender, A.H.F., Ribeiro-Neto, B.A., & da Silva, A.S. (2002a). Debye—Data extraction by example. Data & Knowledge Engineering, 40(2), 121–154.

Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S., & Teixeira, J.S. (2002b). A brief survey of web data extraction tools. SIGMOD Record, 31(2), 84–93.

Lafferty, J.D., McCallum, A., & Pereira, F.C.N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the ICML (pp. 282–289).

Lawrence, S., Giles, C.L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. Computer, 32(6), 67–71.

Lee, D., Kang, J., Mitra, P., Giles, C.L., & On, B.-W. (2007). Are your citations clean? New scenarios and challenges in maintaining digital libraries. To appear in Communications of the ACM, 50(12), 33–38.

Liu, B., Grossman, R., & Zhai, Y. (2003). Mining data records in web pages. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 601–606). New York: ACM Press.

Mesquita, F., da Silva, A.S., de Moura, E.S., Calado, P., & Laender, A.H.F. (2007). Labrador: Efficiently publishing relational databases on the web by using keyword-based query interfaces. Information & Process Management. in press.

Muslea, I., Minton, S., & Knoblock, C.A. (2001). Hierarchical wrapper induction for semistructured information sources. Autonomous Agents and Multi-Agent Systems, 4(1–2), 93–114.

Open Archives Initiative. (2005). The Open Archives Initiative protocol for metadata harvesting. Retrieved October 26, 2005, from http://www.openarchives.org

Paynter, G.W. (2005). Developing practical automatic metadata assignment and evaluation tools for internet resources. In M. Marlino, T. Sumner, & F.M.S. III (Eds.), In Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries (pp. 291–300), Denver, CO. New York: ACM Press.

Peng, F., & McCallum, A. (2006). Information extraction from research papers using conditional random fields. Information & Process Management, 42(4), 963–979.

Reis, D.C., Golgher, P.B., Silva, A.S., & Laender, A.F. (2004). Automatic web news extraction using tree edit distance. In Proceedings of the 13th International Conference on the World Wide Web (pp. 502–511). New York: ACM Press.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. Machine Learning, 34(1–3), 233–272.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. Biometrics, 1(6), 80–83.

Yilmazel, O., Finneran, C.M., & Liddy, E.D. (2004). Metaextract: An NLP system to automatically assign metadata. In Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, Collaboration and Group Work (pp. 241–242).

DOI: 10.1002/asi