cc-IFF: A Cascading Citations Impact Factor Framework for the Automatic Ranking of Research Publications*

Dimitris A. Dervos [1), Thomas Kalkanis [2)
1) Information Technology Dept., T.E.I., P.O. BOX 141, 57400, Sindos, Greece. Email: *dad@it.teithe.gr*
2) Applied Informatics Dept., University of Macedonia, 156 Egnatia Str., P.O. Box 1591,
54006 Thessaloniki, Greece. Email: *pmse0467@uom.gr*

Abstract: *A new framework is proposed for the calculation of impact factor ratings of research publications. Given a collection of research articles, the corresponding citations graph is constructed in the form of a relational table. The impact value is considered at the article level, and is calculated by considering not only the citations made directly to an article, but also citations made to the corresponding citing article(s). In this respect, an improved algorithm is utilized, namely one that traverses all the threads in the citations graph, in an attempt to improve the degree of fairness in assigning credit for the impact value of each one article. When two articles have an equal number of (direct) citations, the one that has triggered more research activity (i.e. its citing articles attract a larger number of citations at subsequent levels in the citations graph) is assigned a higher impact value and, consequently, is ranked to be better.*

Keywords: citation analysis, citations graph, impact factor, research evaluation

1. INTRODUCTION

A citations graph captures the citation indexing inter-relationships for a given collection of research publications (e.g. journal and conference proceedings articles), at a given instance in time. Steve Lawrence, C. Lee Giles, and Kurt Bollacker remark that citation indexing improves scientific communication by[1]:

- revealing relationships between articles,
- drawing attention to important corrections or retractions of published work,
- identifying significant improvements of criticisms of earlier work, and
- helping limit the wasteful duplication of prior research

Citation index based analysis comprises an important stage of the journal and conference evaluation ranking process. The output of the latter is taken into consideration in relation with the tenure and salary levels of researchers and academics, as well as in the funding of research projects[2]. Citation analysis is carried out by utilizing the impact factor metric, introduced by Eugene Garfield, Chairman emeritus of the Institute for Scientific Information (ISI)[3]. Today, the three most widely known systems that implement citation analysis are the Web of Science (WOS)[4], CiteSeer[5], and SCOPUS[6] .

By utilizing the impact factor metric and a number of analogous alternative notions, article collections may be ranked, thus making possible the quantitative measurement of the quality of the hosting scientific journal, or conference[1,2]. However, there exist many conflicting opinions on the validity of the impact factor concept and the way it is implemented in measuring scientific quality[7].

To the best of our knowledge, the quality of an individual article is measured today by considering only the number of its (direct) citations, as well as by means of the impact factor of the journal or conference proceedings it is published with. The higher the impact factor of the (hosting) journal or conference proceedings publication, and the larger the number of citations, the better is the quality of the article in question.

In our opinion, the existing scheme is not fair to articles that deserve more credit for having attracted a large number of citations, despite the fact that they have been published with journals or conference proceedings of a relatively low impact factor value. Also, unfair is the scheme when it comes to evaluating articles that have spurred research activity in their field. A clear indication of the latter is when an article is cited in publications that in turn are heavily cited in subsequent works (i.e. they are not dead-end type research reports).

In this paper, we propose a new framework for the evaluation of the number of citations for which a research publication is given credit. The scheme considers not just the number of citations made directly to the article in question, but also the ones made to the corresponding citing article(s). Our approach clearly involves the (recursive) notion of tracing a number of *cited-to-citing* article pairs, up to a specified depth, along the corresponding threads in the citations graph topology. Thus, our proposal is code-named ‘cc-IFF’, where ‘cc-‘ stands for *cascading citations* and ‘IFF’ stands for *impact factor framework.*

2. The cc-IFF CONCEPT, BY EXAMPLE

In the citations graph presented in Figure 1, a hypothetical collection of ‘articles’ is considered. The articles are labeled with numbers from 1 to 19, and they are

[*] Amended (post-print) version of: D.A. Dervos and T. Kalkanis, *cc-IFF: A Cascading Citations Impact Factor Framework for the Automatic Ranking of Research Publications*, Third IEEE International Workshop on Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Proceedings pp. 668-673, Sofia, Bulgaria, September 5-7, 2005.

assigned to nodes. The directed edges in the graph represent citations made from the corresponding source node (article) to the corresponding target (article). For example, article ‘2’ is seen to cite article ‘1’: the relationship is represented by the arrow originating from node ‘2’ that targets node ‘1’.

Ideally, one would expect the graph in Fig. 1 to be acyclic, by definition: it is not possible to involve cycles, because each citing article is always posterior to the one(s) it cites. This fact would make possible the traversal of the graph by means of a recursive algorithm that is guaranteed to terminate in the general case, even when all of the possible (cascading) citation threads are considered. In practice, this is not always the case; for example, it is possible for a journal preprint to receive a citation from another article that is published at an earlier date than the cited article. Quite possible is also the case of having two published works cite one another. To opt for such cases, it is safer for the recursive algorithm used to be tuned to traverse up to a maximum number of levels along each one thread in the citations graph, since it is not guaranteed to always terminate.

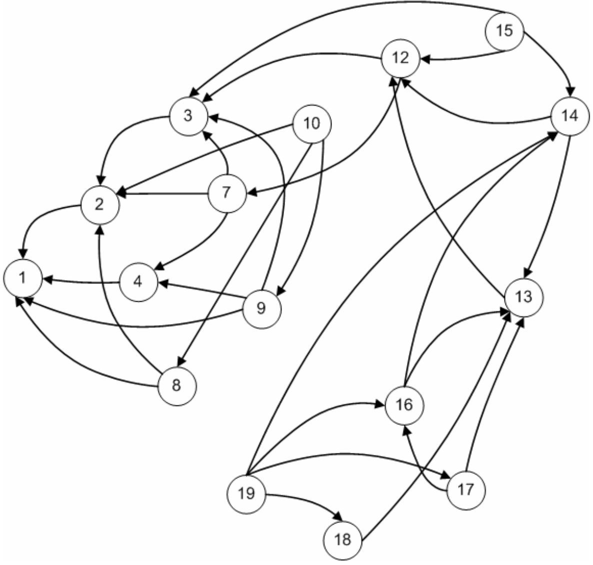

Fig 1 - Citations graph of the hypothetical collection

The information in Fig. 1 may be transferred easily to a relational table, by means of a recursive SQL statement. For the example considered, Table 1 samples on the threads present in Fig.1.

Table 1. Citations graph threads (sample)

CITED	VIA	CITING
. . .	. . .	. . .
01	-02-03-07-12	14
01	**-02-03-07-12**	**15**
01	-02-03-07-12-13	14
01	-02-03-07-12-13	16
. . .	. . .	. . .

For example, the data in the highlighted row of Table 1 read as follows: *article 01 is cited indirectly by article 15*, along the *15-cites-12-that-cites-07-that-cites-03-that-cites-02-that-cites-01* thread of the citations graph.

In the rest of the paper, whenever reference is made to a thread in the citations graph, the article making the topmost citation in the hierarchy (article 15 in the example of the previous paragraph) is called the *source* article. Also, the recipient of the lowermost citation (article 01 in the example) is called the *target* article. We refer to the first generation citations, made directly to an article, as being its *1-gen* citations. In an analogous manner, the 2nd, 3rd,…, nth generation citations, i.e. those made indirectly to an article at subsequent levels in the citations graph, we call *2-gen*, *3-gen*, and *n-gen* citations respectively, *n* being the level number of the source article with respect to the (cited) target article in the citations graph. Thus, for the example case considered in the previous paragraph, article 15 makes a *5-gen* citation on article 01.

3. ISSUES CONSIDERED

Two variations of the cc-IFF algorithm were considered:

- variation-1 calculated the improved citations metric at the *article* level.
- variation-2 calculated the improved citations metric at the (*author*, *article*) level.

The two approaches differed in the way they identified (and excluded from the calculation of the improved citations metric) self-citations. In the case of variation-1, a self-citation occurs when the intersection set result between the authors-lists of the target and source articles is not empty. In variation-2, a self-citation occurs when the name of the author in question appears in the authors-lists of both the source and target articles. It turns out that variation-2 comes closer to the way self-citations are handled in practice and for this reason this is the approach that is considered in the rest of the paper.

During this, initial, stage, no attempt has been made to assign a weight to each citation instance, direct or indirect. One could easily apply a heuristic scheme whereby (say) a *n-gen* citation contributes a $(1/2)^{n-1}$ value to the target item’s citations metric (*n* is effectively the number of edges present in the corresponding citations thread, i.e. $n \geq 1$). The output of the cc-IFF algorithm comes in the form of a medal standings table in athletics, the (*author*, *article*) pairs being ranked in the (*1-gen* citations, *2-gen* citations,…, *n-gen* citations) order. In this respect, the scheme is flexible for implementing a cut-off value on the maximum number of levels to be considered along the typical citations graph thread, as well as for assigning a weight value to each one level.

4. THE TESTBED

In order to apply the cc-IFF principle in practice and to calculate the improved citations metric for a number of real life research publications, the CiteSeer environment was utilized. The main reason for choosing CiteSeer was that a good part of the core database content is publicly available for retrieval in XML format. We say “a good part”, and not “the whole of the core database content”, because the two are extracted from the source repository by applying two separate autonomous citation indexing (ACI) methodologies.

To populate their core database, CiteSeer implement heuristics in parsing the citations listed in the documents of the target collection. In addition, a number of methods are supported for the identification and grouping of the citations to identical articles, ranging from string distance measurements to probabilistic models that identify subfields from the words contained in the citations[1]. On the other hand, the content that is available in XML format comprises the output of machine learning type of processing that extracts open archives initiative (OAI) compliant metadata from the target collection. More specifically, a support vector machine classification-based method is utilized for metadata extraction[8].

The utilization of two methodologies in the construction of the core database and OAI metadata content introduces differences in the two versions of the data content. With regard to author and title parsing, the OAI data is by far the better source, but does not usually include full bibliographic details such as publisher, year of publication, etc., at the present time. Along the same lines, a number of problematic cases were encountered with the OAI metadata used in the current research project, the cases and the way they have been handled, are reported in the next section.

At present, the CiteSeer document repository involves a total of 723,140 electronic articles, focusing primarily on the computer and information science literature. The figure is small when compared to the 14,000 peer-reviewed journal titles of the SCOPUS database, and to the 8,700 peer-reviewed journal titles of the WOS database. However, for a first test of the cc-IFF approach, plus by being freely available to the public, the CiteSeer collection was rated to suffice.

5. DATA PREPARATION AND PROCESSING

In Fig. 2 the typical OAI xml data record is presented, the fields of interest appearing in boldface.

```
<record>
<header>
<identifier>article identifier</identifier>
<datestamp>date retrieved</datestamp>
<setSpec>specifications</setSpec>
</header>
<metadata>
<oai_citeseer:oai_citeseer
            xmlns:oai_citeseer=xml info
            xmlns:dc=xml info
            xml xmlns:xsi=xml info
            xml xsi:schemaLocation=xml info >
  <dc:title>title</dc:title>
  <oai_citeseer:author name=author name>
    <address>author address</address>
            <affiliation>affiliation</affiliation>
  </oai_citeseer:author>
  <dc:subject>subject</dc:subject>
  <dc:description>description</dc:description>
  <dc:contributor>contributor</dc:contributor>
  <dc:publisher>publisher</dc:publisher>
  <dc:date>date published</dc:date>
  <dc:format>format</dc:format>
  <dc:identifier>xml info</dc:identifier>
  <dc:source>xml info</dc:source>
  <dc:language>language</dc:language>
  <oai_citeseer:relation type="References">
    <oai_cs:uri>references</oai_cs:uri>
  </oai_citeseer:relation>
  <oai_citeseer:relation type="Is Referenced By">
    <oai_cs:uri>referenced by</oai_cs:uri>
  </oai_citeseer:relation>
  <dc:rights>rights</dc:rights>
</oai_citeseer:oai_citeseer>
</metadata>
</record>
```

Fig 2 – Typical OAI xml data record

The data in Fig. 2 were processed with the *xml parser* software, developed by Stefan Heymann[9]. For the needs of the research project in question, the fields of interest were the *identifier* and the *title* of each article, its authors-list, publication date, plus the lists of *references-to*, and *referenced-by* article identifiers. The data extracted were used to populate three tables (Citations, Articles, and Authors) of a relational schema created in a MySQL Beta v.5 RDBMS environment (Fig. 3).

```
CREATE TABLE `Citations` (
 `identifier` int(11),
 `isreferencedby` int(11),
 KEY `Index_1` (`identifier`),
 KEY `Index_2` (`isreferencedby`),
 KEY `Index_3` (`identifier`,`isreferencedby`),
 KEY `Index_4` (`isreferencedby`,`identifier`))

CREATE TABLE `Authors` (
 `identifier` int(10),
 `author` varchar(255),
 KEY `Index_1` (`identifier`),
 KEY `Index_2` (`author`),
 KEY `iidentifier` (`identifier`,`author`),
 KEY `Index_4` (`author`,`identifier`))

CREATE TABLE `Articles` (
 `identifier` int(11),
 `title` text
 `date published` date)
```

Fig 3 – OAI metadata relational schema

The data were processed by executing SQL statements directly on the MySQL database interface, as well as by coding in Borland Delphi v.7 that would interface to the database. A total of 574,178 OAI xml article records were processed. With the given collection, five cases that are worth mentioning were identified:

1. There were instances where the *authors* field registered names that do not correspond to physical persons. For example, some article entries had university names included in their authors-lists.
2. Around 49,000 entries were found to register an empty authors-list.

3. In contrast with the CiteSeer core database, the *references* field registered only articles included in the CiteSeer repository.
4. The *referenced by* field was not in agreement with the corresponding field of the core database.
5. Nearly 6,000 entries registered articles that referenced themselves.

Cases 1,2, and 5 are attributed to the ACI algorithm used for the automatic extraction of the metadata (CiteSeer have informed us that a comparison of author data with manually extracted metadata has indicated that the algorithm finds author names perfectly 85% of the time). For the needs of the research project in question, case number 1 possessed no problem, since it was only a small portion of the records that registered such an extra 'noise' data in the *authors* field. Cases 2 and 5 were dealt with by simply removing the problematic article entries. Case number 3 possessed no problem, since our citations graph was restricted to only register articles of the CiteSeer repository, by construction. Finally, case number 4 was dealt with by utilizing the *references* field of the OAI xml data to populate the *isreferencedby* column of the Citations table in the relational database schema of Fig. 3.

In the beginning, the Citations table of the database was measured to register 1,271,898 rows. These rows included problematic entries involving articles from cases 1-5, considered in the previous paragraph. In addition, the table registered instances of special self-citations that should be excluded from the citations graph. More specifically, these were cases of direct (*1-gen*) citations involving articles with identical author-lists. Quite naturally, it is not possible for the target article to act as a bridge in passing the citation from the source article to any one of its authors who might appear in the authors-list in any one of its cited (i.e. predecessor) articles. In this respect, rows that register such special cases of self-citations need be removed from the Citations table.

At this point, it is worth repeating that the cc-IFF framework calculates the improved citation metric for (*author*, *article*) pairs, not for articles. This means that direct citations involving a non-empty authors-lists intersection set result need be registered in the citations graph (consequently: in the Citations table) when either one of the two authors-lists involved is a superset of the other. Thus, it is only the direct citations involving identical authors-lists in both the citing and cited articles that need be excluded from appearing in the citations graph. Other than this, no further restrictions apply to the articles encountered at any one level in-between the source and the target articles of any thread in the graph.

For the testbed database considered, a total of 49,963 *1-gen* citations were found to involve articles with identical authors-lists. Following their removal, as well as the removal of the rows involving articles from cases 1-5, the Citations table was left with 1,065,035 rows (i.e. direct citations).

Next, the improved citations metric (composite) value was calculated up to level number three (i.e. *1-gen*, *2-gen*, and *3-gen*) for each one (*author*, *article*) pair in the database. To speed up processing, the multi-threading algorithm of Fig. 4 was utilized.

```
par_begin
    for each one article
        for each one author in the authors-list
            calculate the 1-gen sum
    for each one article
        for each one author in the authors-list
            calculate the 2-gen sum
    for each one article
        for each one author in the authors-list
            calculate the 3-gen sum
par_end
```

Fig 4 – multi-threading algorithm (pseudo-code of)

Evidently, the algorithm in Fig.4 may easily be extended to calculate *k-gen* ($k > 3$) values.

6. RESULTS

The cc-IFF code was run on an HP ZX5171EA laptop incorporating 512 Mbytes main memory, and an Intel Pentium 3.2 GHz microprocessor. It took nearly an hour and ten minutes to calculate the number of the *1-gen*, *2-gen*, and *3-gen* citations for each one of the 1,065,035 (*author*, *article*) pairs in the database. This was taken to be indicative of the efficiency of the multi-threading algorithm in Fig. 4. Table 2 lists the top 25 (*author*, *article*) entries of the output results obtained.

Table 2. – Top 25 (*author, article*) entries

Author	Title	*1-gen*	*2-gen*	*3-gen*	Year
R.E.Bryant…	Graph-Base…	1301	6830	33308	1986
S.Kirkpatr…	Optimizati…	1147	4949	20734	1983
M.P.Vecchi…	Optimizati…	1147	4949	20734	1983
C.D.Gelatt…	Optimizati…	1147	4949	20734	1983
M.J.Karels…	Congestion…	941	12886	103503	1998
Van Jacobs…	Congestion…	931	12833	103336	1998
A.Pnueli…	Statechart…	892	4391	18150	1987
D.Harel…	Statechart…	881	4360	18018	1987
R.L.Rivest…	A Method f…	861	7184	42528	1978
L.Adleman…	A Method f…	861	7184	42528	1978
A.Shamir…	A Method f…	861	7184	42528	1978
J.K.Ouster…	Tcl and th…	838	5456	29789	1994
Van Jacobs…	Random Ear…	713	6377	39972	1993
S.Ramakris…	Fast Algor…	710	3043	10122	1994
R.Agrawal…	Fast Algor…	708	3037	10108	1994
S.Floyd…	Random Ear…	688	6252	39277	1993
J.Van De W…	Fast Aniso…	680	3222	12669	2002
J.M.Geuseb…	Fast Aniso…	680	3222	12664	2002
A.W.Smeuld…	Fast Aniso…	679	3221	12660	2002
K.E.Schaus…	Active Mes…	677	6541	40462	1992
S.C.Goldst…	Active Mes…	674	6533	40445	1992
T.Von Eick…	Active Mes…	667	6507	40285	1992
D.E.Culler…	Active Mes…	656	6426	40001	1992
T.Imielins…	Mining Ass…	606	5400	28739	1993
S.Deering…	RSVP: A Ne…	605	4433	23519	1993

The entries in Table 2 are sorted first by the *1-gen*, then by the *2-gen*, and then by the *3-gen* value. The *Year* column values were entered manually, by consulting the core CiteSeer database, since the content of the *date published* field of the OAI xml data records (Fig. 2) is not yet fully synchronized with the corresponding field in the core database.

Commenting on the data in Table 2, one notes the entry in the fifth row (*M.J.Karels, Congestion Avoidance and Control*) to have obtained almost twice as many *2-gen* citations, and almost three times as many *3-gen* citations when compared to the topmost entry (*R.E.Bryant, Graph-Based Algorithms for Boolean Function Manipulation*). The way citations are rated and article authors are given credit for today, R.E. Bryant is a clear winner over M.J. Karels for having obtained 1301 next to 941 *1-gen* citations. This is so, despite the fact that R.E. Bryant's article falls behind M.J. Karels' article in the research activity it has spurred (an indicative factor of which may be taken to comprise the numbers of *2-gen* and *3-gen* obtained). Furthermore, M.J. Karels' article was published twelve years after R.E. Bryant's article: 1998 vs. 1986; this is yet one more reason why M.J. Karels should be given more credit for the research interest he has triggered with his article.

Immediately after the (*M.J.Karels, Congestion Avoidance and Control*) row comes the (Van Jacobson, *Congestion Avoidance and Control*) entry. Van Jacobson and M.J. Karels are co-authors of the *Congestion Avoidance and Control*) article. Still, the former is falling slightly behind the latter on the number of *1-gen, 2-gen*, and *3-gen* citations obtained; this is attributed to a larger number of self-citations to the paper in question being present for Van Jacobson.

Fig. 5 provides a concise view on how *1-gen, 2-gen*, and *3-gen* citations go for the top 50 (*author, article*) pairs in the CiteSeer database. On the horizontal axis, the 50 (*author, article*) pairs are encoded in accordance with their ranked position in the output. This means that the (*R.E.Bryant, Graph-Based Algorithms for Boolean Function Manipulation*) entry is labeled to be entry number 1, and so on. The vertical axis lists the number of citations made.

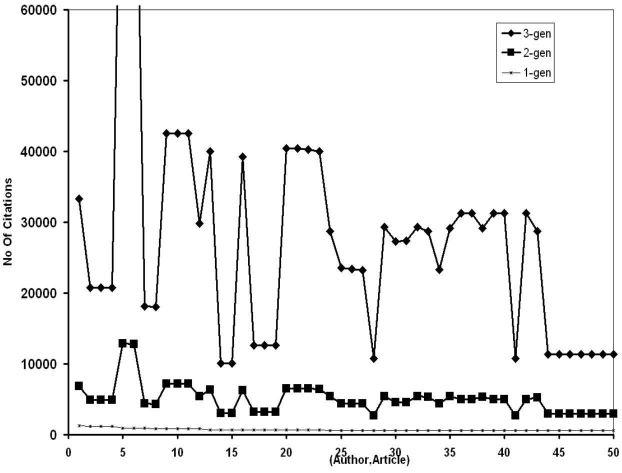

Fig 5 – Top 50 *1-gen, 2-gen*, and *3-gen* citations

Observing the graph in Fig. 5, one can easily identify the (*author, article*) pairs that fall behind in the number of *2-gen* and *3-gen* citations obtained: they are the ones labeled as numbers 1,2,3,4,7,8,14,15,17,18,19,28, and 41. This one alone not being a reason for reaching a conclusion on the impact a published article has made in promoting science, the information obtained may probably be utilized for the (*author, article*) pairs that represent the peaks in Fig 5 to be given more credit for the research interest they have triggered.

7. CONCLUSION AND FURTHER WORK

A new framework for the calculation of improved impact value ratings for research publications is proposed. The scheme involves the calculation of the number of citations by considering not just the *1-gen* ones, but also those made indirectly in a (finite) number of subsequent levels of the threaded hierarchies in the citations graph. In the present report, the first three levels are taken into consideration.

In the next stages of the work, the cc-IFF algorithm will be improved to identify instances whereby a source article cites the same target article along more than one citations thread in the graph. For example, considering the citations graph of Fig. 1, article '9' cites article '1' along three threads: (a) 9→1, 9→4→1, and 9→3→2→1. For article '1', the scheme involves one *1-gen* and two *2-gen* citations and deserves to be granted a higher impact factor rating from, say, the case where the two *2-gen* citations were to originate from two (as opposed to one) source articles.

Last but not least, the application of the cc-IFF algorithm in a real life situation assumes the presence of a name disambiguation data pre-processing stage[10]. This is one of the issues to be considered next in our research project.

8. ACKNOWLEDGEMENTS

The authors wish to thank Mr. Isaac Council, CiteSeer.IST administrator, and Mr. Levent Bolelli of CiteSeer for their prompt response and for the clarifying answers they have provided on issues relating to the differences in the content of the OAI metadata and the core CiteSeer database. Their assistance is deeply appreciated.

Special thanks are due to Dr. Nikolaos Samaras, Lecturer at the Applied Informatics Department of the University of Macedonia, Thessaloniki, Greece, for a number of mind stimulating discussions on the applicability of the cc-IFF framework, as well as for having proposed the multi-threading algorithm the utilization of which has improved the performance of the *1-gen, 2-gen*, and *3-gen* citations extraction processing stage.

9. REFERENCES

[1] S. Lawrence, C.L. Giles, K. Bollacker, Digital Libraries and Autonomous Citation Indexing, *IEEE Computer Magazine*, 32 (6) (1999), p. 67-71

[2] A. Sidiropoulos, Y. Manolopoulos, A new Perspective to Automatically Rank Scientific Conferences using Digital Libraries, *Information Processing and Management*, 41 (2) (2005), p. 289-312

[3] E. Garfield, Journal Impact Factor: a brief review, *Canadian Medical Association Journal*, 161 (8) (1999), *http://www.garfield.library.upenn.edu/impactfactor.htm*

[4] H. Atkins, The ISI® Web of Science® – Links and Electronic Journals, *D-Lib Magazine*, 5 (9) (1999), *http://www.dlib.org/dlib/september99/atkins/09atkins.html*

[5] C. L. Giles, K. Bollacker, S. Lawrence, CiteSeer: An Automatic Citation Indexing System, *Digital Libraries 98-The Third ACM Conference on Digital Libraries Proceedings*, (1998), p. 89-98

[6] SCOPUS, *http://www.info.scopus.com*

[7] C. Hoeffel, Journal Impact Factors [letter], *Allergy*, 53 (1998), p. 1225

[8] H. Han, C.L. Giles, E. Manavoglu, H. Zha, Z. Zha, E.A. Fox, Automatic metadata extraction using support vector machine, *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp 37-48, May, 2003.

[9] *Xml Parser*, available at: *http://www.destructor.de*

[10] H. Han, L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis, Two supervised learning approaches for name disambiguation in author citations, *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, (2004), p. 296-305

Thessaloniki, Greece, 5 April, 2006