Annales Univ. Sci. Budapest., Sect. Comp. 31 (2009) 43-56

REFERENCE EXTRACTION AND COAUTHORSHIP VISUALIZATION OF SEMI-STRUCTURED BIBLIOGRAPHIC DATA

G. Dražić, D. Dobrić, M. Radovanović and M. Ivanović

(Novi Sad, Serbia)

Abstract. This paper describes the steps taken to transform a semi-structured collection of documents containing bibliographic references of researchers from the Serbian province of Vojvodina into an information retrieval service which permits explicit visualization of publication coauthorhips. The process of information extraction consisting of reference recognition and coauthorship detection is presented first, together with an experimental evaluation on a representative subset of the data which demonstrates good values of precision and recall of extraction. Then, an overview of a program is given, which provides services for search and visualization of bibliographic data collected from the semi-structured source. Examples of program usage demonstrate how collaboration of researchers and organizations may be analyzed using the visualization functionalities of the software. Besides (co)authorships, the data collection contains other interesting information which may be utilized for social network analysis of Vojvodinian researchers and organizations in future work.

1. Introduction

Many educational and research institutions are situated in Vojvodina, the northern province of Serbia, covering almost every field of science. Most of these institutions operate under the umbrella of the University of Novi Sad. In 2004, the Provincial Secretariat for Science and Technological Development of Vojvodina started gathering data from researchers employed at the institutions, by having every researcher fill

in a form provided as an MS Word document. Among other data, the forms required researchers to specify complete references of all authored publications. The gathered information was made available in unmodified form on the Web site of the Secretariat: `apv-nauka.ns.ac.yu/`. Notable properties of the data, at this stage, are its semi-structured nature, incompleteness (unfortunately, many researchers did not submit their forms) and diversity of approaches to giving references, permitted by information being entered into the forms in free text format.

This paper describes the process of extraction of references and coauthorship relations from the collection of documents describing Vojvodinian researchers, thus transforming semi-structured data into a fully structured database. It extends our previous work described in (Radovanović et al. [10]) with a formal experimental evaluation of the accuracy of extraction and presents several improvements initiated by the evaluation. Then, an information retrieval system is presented which allows the results of queries over the database to be visualized as collaboration graphs expressing co-authorship of papers between researchers or organizations. At this stage, it mimics the collaboration graph functionality of the IST World portal (Jörg et al. [5, 6]; Radovanović et al. [10]), with the added ability to expand query results with direct and indirect "neighbors," down to an arbitrary level. The motivation for building a stand-alone visualizer for our data lies in the presence of many specific pieces of information which can not be utilized by the general database schema of IST World based on (CERIF [2]). We plan to expand the visualizer to make use of this information and provide data-specific functionalities for querying, visualization and analysis.

The rest of the paper is organized as follows. Section 2 describes the initial data, the extraction process and the database model to which the extracted information is exported. Section 3 overviews the design of the information retrieval and visualization application that works with the extracted information and provides some examples of its use and possibilities for analysis of coauthorships. The last section presents the conclusions and guidelines for possible future work.

2. Information extraction

This section presents the steps taken to extract information about references contained in the collection of semi-structured documents providing a research bibliography. After describing the data in Section 2.1, the process of reference recognition and coauthorship detection is presented in Section 2.2, together with evaluation of their performance on a representative subset of the data. Section 2.3 outlines the database model to which the extracted data is exported.

2.1. Data description

At the time of writing, the collection of documents (with the last update on July 6, 2006) includes 2,278 researchers from 60 institutions. Despite the large number of entries considering the size of the Vojvodinian region, the collection is still in early stages of development. Many researchers have not yet submitted their data and some information in the collection appears to be out of date, which should be remedied in part by the planned future updates of the database. The number of existing entries in the collection still made the task of manually extracting bibliographical data infeasible. We resorted to programming an extractor in Java which, at this time, is able to automatically isolate every researcher's name, affiliation and list of references, and save the data in various formats. Furthermore, the extractor compares references among different authors, detecting coauthorships among researchers who are included in the collection.

The form to be filled by every researcher consists of a sequence of tables starting with basic data (name, year of birth, etc.), continuing with the tables corresponding to publication types as prescribed by the Serbian Ministry of Science and Environmental Protection. Publication types are labeled by a code of the form Rxx, where xx is a two-digit number. The codes of interest have the first digit in $\{1, 2, 5, 6, 7\}$, which corresponds to published papers and book chapters, and excludes technical solutions (3) and patents (4). A sample entry is shown in Table 1. We observed that within the tables, the references were usually entered enclosed in isolated paragraphs or numbered lists. The collection includes references written in more than five natural languages, the most prominent being Serbian, English, Hungarian, Slovak and Romanian.

Table 1. Example entry in the form. R52 corresponds to papers published in international journals of category 2

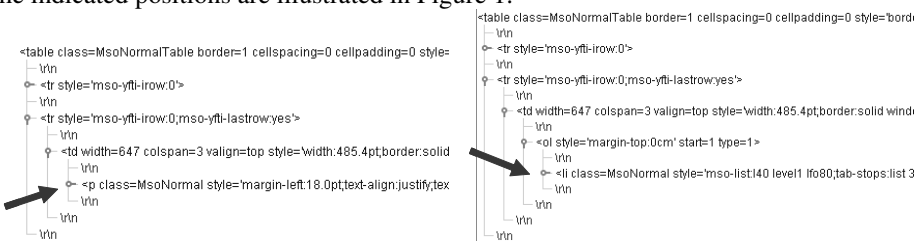
Spisak rezultata R52 - Rad u časopisu međunarodnog značaja. Međunarodne časopise i druge navode rangirati (koeficijent R) prema Science Citation *Index-u (Journal Citation Report) odnosno prema kategorizaciji radova, verifikovanih od strane odbora Ministarstva.	**Broj**	**10**
1. Bađonski, M., Ivanović, M., and **Budimac, Z.**, Software Specification using LASS. In *Proc. of ASIAN '97* (Kathmandu, Nepal), Shyamasundar, R. K. and Ueda, K, eds., Lecture Notes in Computer Science vol. 1345, Springer Verlag, Berlin, 1997, pp. 375-376. 2. **Budimac, Z.** Mašulović, D., Linda as an Abstract Data Type for Concurrent Programming, *Novi Sad J. Math* 28 (1998) 2, 173-186 (Publisher: Faculty of Science, University of Novi Sad, Novi Sad). . . .		

2.2. The extractor

Version 2.0.2 of the extractor is able to isolate a total of 101,672 bibliographic units from current data and detect 24,262 duplicate references (which correspond to coauthorships – a paper appearing in n researchers' forms can have a maximum of $n-1$ detected

duplicates). This makes the total number of references in the database 77,410. The researchers' names and affiliations are extracted from the HTML page on the Web site of the Secretariat for Science in a straightforward fashion, which left the biggest challenge in processing the reference data from MS Word documents.

Reference recognition. From the limited number of options for accessing the content of MS Word documents from outside programs, we found it most convenient to bulk convert all documents to HTML format via a Word macro, and do all actual extraction from HTML. The HTMLParser open source library is used to process the generated HTML files and isolate the DOM trees of <TABLE> tags corresponding to tables containing the references of interest, as described in Section 2.1. Further extraction of references is done using the following scheme: since it was observed that isolated paragraphs and numbered lists in Word documents correspond to <P> and list tags in generated HTML, the references were "read out" from fixed positions in the DOM trees of <TABLE> tags, taking into account the two above possibilities. The DOM trees with the indicated positions are illustrated in Figure 1.



(a) (b)

Figure 1. Possible positions of references in the HTML DOM tree

Somewhat surprisingly, this simple scheme turned out to be rather effective at retrieving strings containing valid references. After observing the extracted references, we removed from the collection the 59 forms which were obviously not parsed correctly within this scheme (the references were either divided up into several parts or lumped together). The forms which were filled in using the Cyrillic alphabet were also removed (62 in total), since it was elected to leave the conversion of Cyrillic letters for a later date. From the remaining forms, the parser could not correctly process 444 tables (out of a total of 39,688), which roughly corresponds to 17 whole forms. All this amounts to 138 unprocessed forms, putting the upper bound on recall to around 94%.

In order to evaluate more precisely the success of reference recognition, we examined the results of extraction on 42 forms submitted by researchers from the Department of Mathematics and Informatics. We counted the *true positives* (TP), which in this context is the number of extracted strings that truly are bibliographic references, the *false positives* (FP) – the number of strings which are only partial references or not references at all, and the real number of references in the submitted forms. The classical measures from information retrieval, *precision* and *recall*, may now be expressed as

$\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and $\mathrm{TP}/\mathrm{Real}$, respectively.

The results of the evaluation, summarized in Table 2, revealed a recall of only 62.47%. It was immediately evident that this was due to some forms being filled in an unexpected manner – references have sometimes been written in the header rows of the tables instead of the provided second row. After removing 9 such forms, recall jumped to 89.21%, meaning that a simple adjustment of the parsing scheme should considerably raise recall. On the other hand, precision was determined to be 97.92%, exceeding our initial estimate of 97% (Radovanović et al. 2006). Also, it was observed that some tables were being selected for reference extraction when they should not have been, that way damaging precision.

Table 2. Evaluation of reference recognition for extractor v2.0.2

	TP	**FP**	**Real**	**Precision**	**Recall**
True estimate	1082	23	1732	97.92%	62.47%
9 forms removed	1033	21	1158	98.01%	89.21%

Based on the above observations, in extractor v2.0.3 we introduced several enhancements:

- When references are inclosed in <P> tags, if the content of the <P> tag does not begin with an ordinal number it is concatenated with the previous reference. This fix ensured that references are no longer divided up;
- The first row of tables is now being searched for references;
- Selection of tables was made more accurate.

Table 3 summarizes the evaluation of the new version of the extractor on identical data. It can be seen that both precision and recall are considerably improved. The reason for recall not being closer to 100% lies in specific parsing issues within certain forms and tables. We decided not to address these details in order to leave the evaluation unbiased: introducing fixes that solve problems specific to the chosen evaluation sample would have led to "overfitting" and producing overly optimistic estimates of precision and recall. The enhancements that were introduced to the extractor are general, in the sense of pertaining to all forms, not just the chosen evaluation dataset.

Table 3. Evaluation of reference recognition for extractor v2.0.3

TP	**FP**	**Real**	**Precision**	**Recall**
1580	5	1732	99.68%	91.22%

In summary, version 2.0.3 of the extractor isolates 110,394 references (about 9,000

more than v2.0.2) and detects 30,822 duplicates (about 6,500 more), making the total number of references in the database 79,572. Detection of duplicates is discussed next.

Coauthorship detection. In order to calculate the similarity of two references, with the intention to determine a coauthorship relation, the extractor uses an optimized version of the algorithm described by White ([11]), which calculates the value of the Tanimoto similarity metric over the space of character 2-grams. The algorithm computes the ratio between the number of shared 2-grams (letter pairs) and the number of all 2-grams in both strings, disregarding whitespace, punctuation marks and capitalization. The ratio is multiplied by two to keep the resulting measure between 0 and 1. More information on string similarity metrics is available in Chapman ([3]), Kohonen ([8]) and Cohen et al. ([4]), while character n-grams are discussed by Cavnar and Trenkle ([1]), and Lodhi et al. ([9]).

The reason for using 2-grams instead of, for instance, whole words, lies in the observed "dirtiness" of manually entered reference data: typographic errors, different or missing information, various referencing conventions used (resulting in different ordering of reference information), etc. After parsing a researcher's form and extracting a list of references, every reference is compared to all references already in the database which contain the researcher's last name (actually, its first word), retrieved using a maintained index. If the best match of a given reference does not exceed a predetermined similarity threshold (set at 0.63 after examining several test cases), the reference is entered as a new one into the database. Otherwise, a coauthorship relation is established, and the entry for the currently processed reference of the researcher is set to point to the reference already in the database.

Table 4. Evaluation of coauthorship detection for extractor v2.0.3

TP	FP	Real	Real (no lang.)	Precision	Recall	Recall (no lang.)
583	26	625	612	95.73%	93.28%	95.26%

Evaluation of coauthorship detection conducted on the same data as the evaluation of reference recognition is summarized in Table 4. Numbers for true positives, false positives and the real count actually represent authorship relations of researchers to multi-authored papers - papers with two detected authors are counted twice, with three authors three times, etc. Precision is 95.73%, lowered from the perfect score by wrong assignments of one author to a two-author paper, which happened among authors with same last names and similar scientific interests. Undetected coauthorships arose mainly for three reasons: (1) one author was supplying much less information within a reference than another, that way lowering the calculated string similarity, (2) an author changed her name, or (3) authors wrote references in different languages. Information which could help solve case (2) was usually not available within the forms. Since situation (3) requires a sophisticated solution, recall was calculated separately for the two cases when

different language references are considered equal and not equal in reality, resulting in recall values of 93.28% and 95.26%, respectively.

2.3. The database model

Figure 2 shows the conceptual database model to which the extracted data is converted. As it can be seen, the main entities are "Institution," "Researcher" and "Publication." There are two more tables, "res_pub" and "inst_pub," which associate researchers

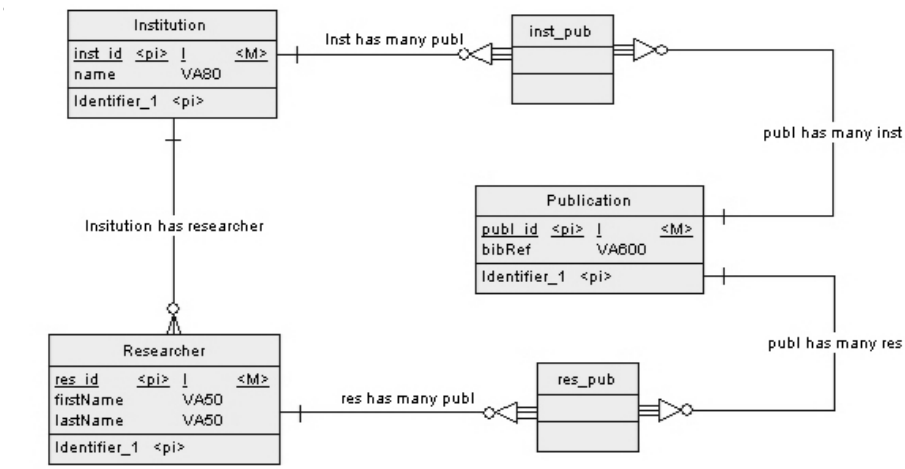


Figure 2. The database model

This simple database model proved sufficient for retrieval and visualization of extracted information, as described in the next section. Since researcher's forms provide more data than is currently being extracted, the database model will be extended with new fields and relations together with the development of new functionalities of the visualizer software.

3. The visualizer

This section will be dedicated to describing a program for visualizing the connectivity of scientists (or institutions) based on authored publications. The view is conceived in the form of a graph where the nodes represent scientists (or institutions) and the lines

and their thickness represent the number of collaborative publications where the scientists appear as coauthors (or in the case of an institution, the number of collaborative publications among scientists, employed in these institutions). The following sections will be dealing with the design specification (Section 3.1), as well as several test examples (Section 3.2).

3.1. Design specification

If one takes a closer look at the application model, three functional units can be noticed:

1. Package org.bean which contains classes representing the database model;
2. Package org.manager which contains a class implementing "the business logic" (scans and loading of data from files);
3. Package org.jgraph contains a class representing the user interface of the application.

The view of this package in Eclipse's package explorer is given in Figure 3. A description of the packages follows.

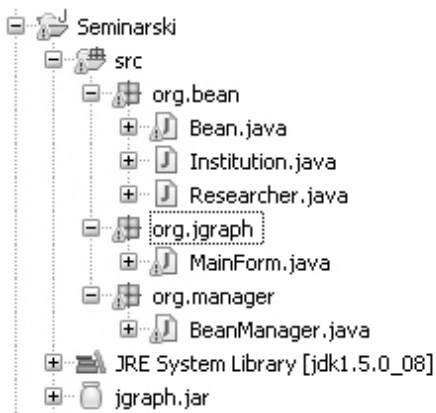


Figure 3. Application overview in Eclipse's package explorer

The org.bean package. This package contains classes representing the model of data being loaded from the file. There are three classes in this model:

- The Bean class is an upper class for other bean classes and it defines two attributes: the bean's ID (all entities in this database have it) and the list of all publications for this respective bean (be it an institution or a researcher);

- The Institution class inherits the Bean class and adds an attribute for the name of an institution;
- The Researcher class inherits the Bean class and adds attributes for the name and surname of a researcher as well as the attribute for the ID of the institution where the researcher works.

The org.manager package. This package contains only one class BeanManager, performing the business logic of the application. During the making of instances for this class, the names of files from which the data will be read are proceeded to this class. The loaded data is saved in beans during the execution of the program. The instances of the beans are found in hash maps which are the attributes of the BeanManager class. Beans are kept in hash maps in order to achieve indexing and provide a faster scan. The placement of beans in hash maps is done in the following manner: by forming pairs [key, value] where the key is the name or surname, and the value is a list of surnames or names corresponding to the key (if the key is the name, then the value is the list of researchers' surnames with this name, and when the key is a surname then the value is the list of researchers' names with that surname).

The indexation of the data belonging to an institution is not based on keys. This data is directly placed in a list where the ID and the name of the institution are retained. The number of institutions is far smaller than the number of researchers, therefore institutions are easier for scanning. The scanning is based on names or just a part of the name of an institution. It is allowed to insert special symbols like "" in keywords for scanning which means that the exact name between these symbols would be tracked.

With researcher queries, quotation marks are taken into account when calculating the result. For example, if the keywords are John Smith, the result of the scanning method will be the list of researchers with the name John or surname Smith, but if the keywords used are "John Smith" the result of scanning will only be a researcher by the name of John Smith (if he exists). It should be mentioned that the scanner is not case sensitive. The query is scanned between empty spaces (blanks). The results are found for every single word, or, in case when keywords are between quotation marks, the intersection of results is being sought. If there are no quotation marks the union of results is performed. If there is more than one keyword used for scanning, it might happen that results satisfying more requirements are obtained. While retrieving these kind of results, duplicates are removed; duplicates being beans found more than once in the list of results. The items also being removed are those beans having a smaller number of connections to other beans than the minimum (which is also a parameter of the method used for scanning).

Neighbors. One more option has been added as a part of the search, and that is connecting with the "neighbors." A neighbor is a researcher (an institution) working on the same publication with the one found during the search. There is also an option to set the number of levels and therefore decide how far the search will go (in that case the neighbors' neighbors are being searched for, and that process can be continued depending on

the number of levels set). Since this kind of search gives back a large number of results it is possible to set a maximum number of results and by doing this shorten the time of the search. It should be mentioned that the search applied to institutions is the same as the one for researchers. The same method is used, the only difference being separate hash tables which are used depending whether researchers or institutions are being searched for.

A short description of how this search works follows. Results are found for all the given keywords and loaded to the result list. The neighbors for all results are now being searched for and added to the list. That is the first level. If more than one level is specified, the results from the first level are used and their neighbors are being searched for and the results added to the list. The process repeats in the same manner for each following level. When placing the found results in the list, a check is performed to see if the number of results is larger than the maximum number of results. If that turns out to be the case, the search is halted.

The org.jgraph package. This package contains the MainForm class which represents the user interface for this application. It contains the main method enabling the application to be run when this class is started. When instancing of this class is performed, the instancing of the BeanManager class is called for, and a file path of data which are used for searching is set as a parameter for this class. After that comes the creating of visual objects and their preview for the user, i.e. the preview of the user interface.

3.2. Examples of coauthorship visualization

This section describes an example collaboration analysis of Vojvodinian researchers and organizations. Following are two examples of how to use the application.

Example 1. The results will be shown here to exemplify what happens when a keyword "fakultet" is typed into a search field for institutions, with the minimum number of connections being 20. The graph with the results is shown in Figure 4, representing only those faculties which have more than 20 coauthorships.

For instance, is can be seen that the Faculty of Science (*Prirodno-matematički fakultet*) has strong ties with the Faculty of Agriculture (*Poljoprivredni fakultet*) through its Department of Biology, and also because of many graduates of the former being employed by the Faculty of Agriculture. The Faculty of Science also collaborates strongly with the Faculty of Technical Sciences (*Fakultet tehničkih nauka*) via its Department of Physics and the Department of Mathematics and Informatics; with the Faculty of Technology (*Tehnološki fakultet*) through its Department of Chemistry; and with the Faculty of Medicine through the Departments of Biology and Chemistry. The most surprising link on the diagram, between Faculties of Medicine and Philosophy, upon closer inspection turned out to be due to an error in the original data: the faculties employ two

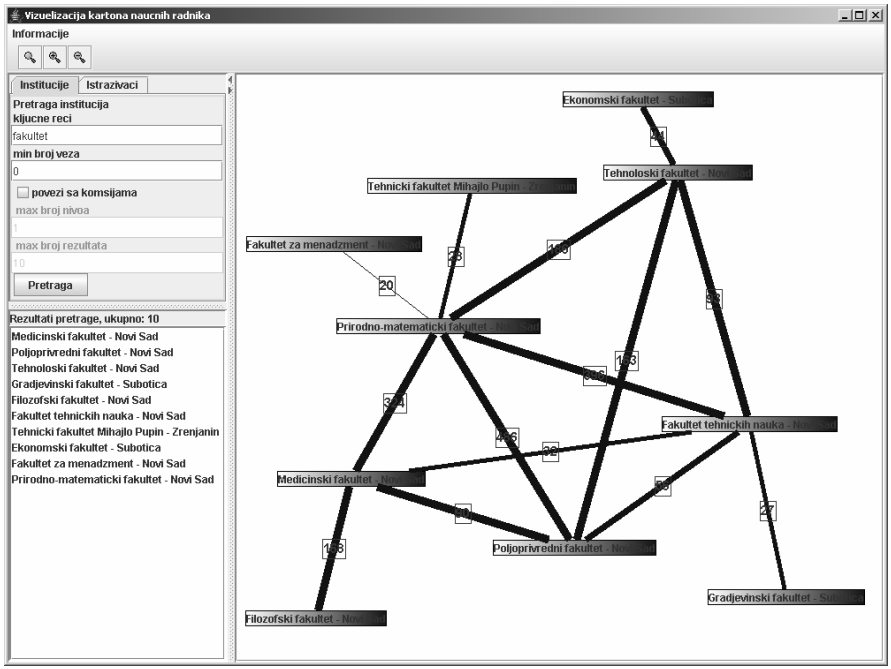


Figure 4. Search results for keyword “fakultet”

different researchers with the same first and last name (*Slobodan Pavlović*), and in the collection they were mistakenly represented by identical forms, resulting in the extractor perfectly matching all 148 publications.

Example 2. This example shows the graph with the results when the keywords “surla dolinka tepavcevic” are typed into the search field for researchers, with the minimum number of connections being 0, and all the connections with all the neighbors up to level 3. The data for this query is restricted to the Department for Mathematics and Informatics (DMI) of the Faculty of Science, University of Novi Sad. The graph in Fig. 5 shows all extracted collaboration between members of the Department who submitted their data. Note that graph layout was manually corrected – we need to do more research on effective drawing of graphs of this type and semantics. Currently a simple, mostly randomized algorithm is used, which does not give particularly good results in the general case.

The strong cooperations between professors Zoran Budimac and Mirjana Ivanović, and professors Miloš Racković and Dušan Surla, represent the “backbones” of the two Informatics chairs at the Department – the Chair of Computer Science and the Chair of

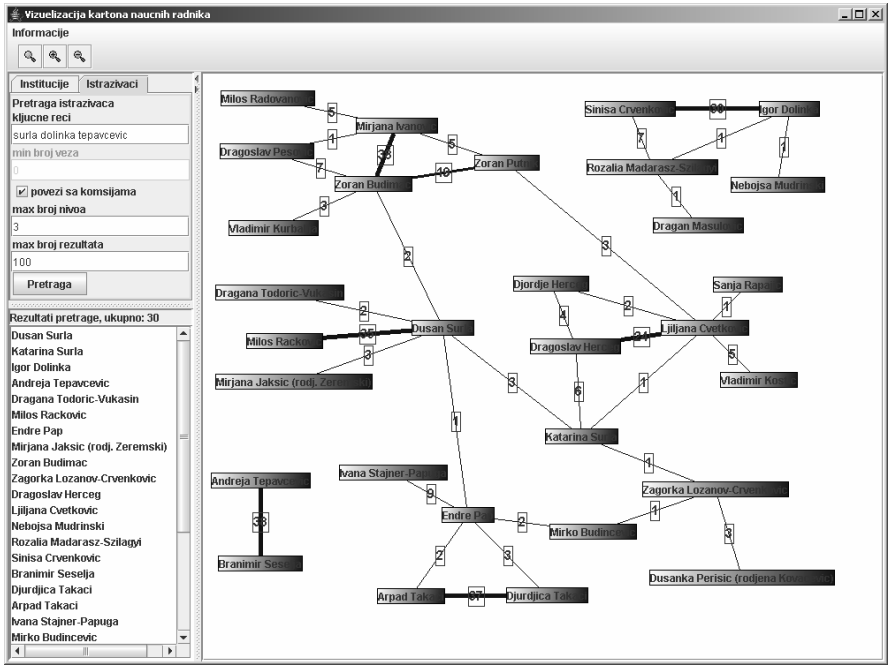


Figure 5. Members of the Department of Mathematics and Informatics

Information Systems, respectively. All other members of both chairs collaborate directly with at least one member of the backbone, and inter-chair cooperation is expressed by two joint papers of Zoran Budimac and Dušan Surla.

The Mathematics chairs, which form the majority of the Department, are also clearly visible in the diagram. The Chair of Numerical Mathematics is represented by prof. Ljiljana Cvetković, prof. Dragoslav Herceg, prof. Katarina Surla and Vladimir Kostić; the Chair of General Algebra and Theoretical Computer Science by profs. Siniša Crvenkovic, Igor Dolinka and Rozália Madarász-Szilágyi; the Chair of Applied Analysis by profs. Endre Pap, Ivana Štajner-Papuga, Đurđica and Arpad Takači; the Chair of Mathematical Analysis, Probability and Diff. Equations by profs. Mirko Budinčević, Dušanka Perišić and Zagorka Lozanov-Crvenković; and the Chair of Applied Algebra by profs. Branimir Šešelja and Andreja Tepavčević. Extraction errors (i.e. lower recall) are only partly to blame for the fact that many members of the Department are missing from the diagram – a number of senior researchers have not submitted their data to the Secretariat for Science, which resulted not only in their exclusion from the diagram, but also in the exclusion of many young researchers who have a low number of publications, coauthored only by their mentors.

4. Conclusions and future work

With the current state of the data regarding its incompleteness, and imperfect precision and recall scores of extraction, the results of information retrieval and visualization of coauthorship cannot be considered 100% true and reliable. Despite this, the relationships that *are* observed between organizations and researchers do comply with our general picture of Vojvodinian research, suggesting that overall precision and recall of extraction are not far from the estimates obtained on the evaluation dataset.

Currently, the extractor processes only whole references, with no attempts to isolate the author list, title, journal or conference name, publication date and similar information. Work is currently being done in this direction, and is made difficult by the variety of used referencing conventions and languages in the forms. If successful, this would allow expressing many other relations beside coauthorship, e.g. being in the same conference/journal issue, same conference stream/journal, or similar conferences/journals (Klink et al. 2006). Another area for exploration is a more comprehensive study for tuning the similarity threshold, and investigating different similarity measures like the *cosine* in different spaces, not only in the n-gram space. Improving reference recognition by implementing more parsing schemes and Cyrillic letter conversion is also on the agenda, as well as experimenting with different graph layout algorithms.

The forms filled in by researchers contain more information than is extracted and used by the current versions of the presented software. Additional information includes: scientific fields of interest, working positions, authored theses and textbooks, patents, thesis mentorships, reviewing activity, etc. All this information may be utilized to infer and visualize many interesting relationships other than (co)authorship. This was the primary motivation for implementing a custom information retrieval and visualization application, which we plan to further extend and develop into a comprehensive system for social network analysis of researchers and organizations from Vojvodina.

References

[1] **Cavnar W. B. and Trenkle J. M.,** N-gram-based text categorization. *Proceedings of SDAIR94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, 1994,* 161–175,

[2] *CERIF: the Common European Research Information Format*, 2000, `http://cordis.europa.eu/cerif/`

[3] **Chapman S.,** *String similarity metrics for information integration*, 2007. http://www.dcs.shef.ac.uk/ sam/stringmetrics.html

[4] **Cohen W. W., Ravikumar P. and Fienberg S. E.**, A comparison of string distance metrics for name-matching tasks, *Proceedings of IJCAI03 Workshop on Information Integration on the Web (IIWeb03), Acapulco, Mexico, 2003,* 73-78.

[5] **Jörg B., Jermol M., Uszkoreit H., Grobelnik M. and Ferlež J.**, Analytic information services for the European research area, *Proceedings of eChallenges e-2006 Conference, Barcelona, 2006,* 1367-1374.

[6] **Jörg B., Ferlež J., Grabczewski E. and Jermol M.**, IST World: European RTD information and service portal, *Proceedings of CRIS06, 8th International Conference on Current Research Information Systems: Enabling Interaction and Quality: Beyond the Hanseatic League, Norway, 2006,* 131-140.

[7] **Klink, S. et al.** Analysing social networks within bibliographical data, *Proceedings of DEXA06, 17th International Conference on Database and Expert Systems Applications*, LNCS **4080**, Springer Verlag, 2006, 489–498.

[8] **Kohonen, T.**, *Self-organizing maps*, 3rd edn., Springer Verlag, 2001.

[9] **Lodhi H., Saunders C., Shawe-Taylor J., Cristianini N. and Watkins C.**, Instance-based learning algorithms, *Journal of Machine Learning Research*, **2** (2002), 419-444.

[10] **Radovanović M., Ferlež J., Mladenić D., Grobelnik M. and Ivanović M.**, Extending the IST-World database with Serbian research publications, *Proceedings of IS2006, 9th International Multiconference on Information Society, Ljubljana, Slovenia, 2006,* 251–254.

[11] **White S.** *How to strike a match*, 2004. http:// www.devarticles.com/c/a/Development-Cycles/How-to-Strike-a-Match/

G. Dražić, D. Dobrić, M. Radovanović and M. Ivanović
Department of Mathematics and Informatics
Faculty of Science
University of Novi Sad
Trg D. Obradovića 4.
21000 Novi Sad, Serbia
drazag@neobee.net, ddragand@gmail.com, {radacha,mira}@im.ns.ac.yu