Data Mining the DBLP Computer Science Bibliography

Linus Dietz
linus.dietz@uni-bamberg.de

Abstract—The DBLP COMPUTER SCIENCE BIBLIOGRAPHY is a web service that provides information about publications in computer science and their authors. Although incomplete, it is an important tool for researchers in this area with over 2.5 million publication records in the database. Due to the OPENDATA license it is possible to work with the provided XML-files and create new analyses of the database.
Using common data mining techniques I explored the database and aggregated the data focusing on the visualisation of coauthorships. Furthermore I created a web interface for custom user requests.

1 INTRODUCTION

The DBLP COMPUTER SCIENCE BIBLIOGRAPHY is one of the largest databases of publications and authors in computer science and thus is a fruitful tool for researchers in this area. Since its beginnings in the 1980s, it covers records of books and articles of substantial journals and conferences [1].

The database can be accessed in various ways, most prominently with the default web interface http://dblp.uni-trier.de, however due to the liberal OPEN DATA COMMONS license other customized web services are available for advanced searches and use cases:

- **CompleteSearch** [2] is a DBLP mirror with extended search capabilities.
- **FacetedDBLP** [3] is another search engine for DBLP enriched with additional metadata and an improved search interface.
- **FreeSearch** [4] is a meta search engine for DBLP, TIBKat, CiteSeer and BibSonomy.

My aim in this work is to build a novel web service using the DBLP database with a narrow focus: Exploring and visualizing communities of authors based on coauthorships in publications.

In the following section I present my approach and the technologies used. In Section 3 I give some descriptive statistics of the relevant records in the DBLP database and I visualize the coauthorship graphs of an example author in Section 4. In Section 5 I describe the functionality of the web service and I draw my conclusions in Section 6.

2 METHODS

Following the CRISP-DM [5] standard, the work on this project was divided into the following phases:

- **Business Unterstanding** In the first phase I made myself familiar with the work of Michael Ley on the DBLP database and the importance it has for research in computer science.
- **Data Understanding** The DBLP dataset is available as a 1.3GB XML file with an corresponding DTD file. Due to its size, it is – in terms of memory – infeasible to build a DOM-TREE. Thus I used the event-driven SAX PARSER in the PYTHON programming language.
  The DBLP database is structured as following: On the high level it contains publication records sorted by type. Possible values are `article`, `book`, `inproceedings`, to name some.
  These records are enriched with additional information: The `title`, `authors` and further information about the publication,

depending on the type. In any case sufficient information is given to be able to generate a BIBTEX file.

- **Data Preparation** As described before, the data is already in a structured, well readable format. Since the parsing of the original XML file usually takes about seven minutes on a standard laptop computer and this file is updated frequently, no additional cleaning of the raw data was done.
- **Modeling** In this step the XML database was read in and several PYTHON data structures were built up, most importantly the coauthors index shown in Listing 1.

```
coauthorsDB :: dict(author ::
   String, (dict(coauthors ::
   String, number of mutual
   publications :: Int))
```

Listing 1. Coauthor Index Data Structure Type

This nested map allows fast access to the authors and the list of their coauthors together with the number of papers they published together. This data is passed to GRAPHVIZ which generates the visualisation as described in Section 4.
The other data structure is a simple map from authors to their publication count.

```
authorPubCount :: dict(author ::
   String, publication count :: Int
   )
```

Listing 2. Publication Count Data Structure Type

The visualisation of the graphs was tweaked a lot to find the optimal GRAPHVIZ settings. As a tradeof between runtime and visualisation quality I decided to use the SFDP algorithm with the "`overlap=false`" and "`splines=spline`" (for the basic coauthors graph) and "`splines=curved`" attributes for the coauthors of coauthors graph.

- **Evaluation** The correctness of the parsing was double checked with the statistics section of the DBLP webpage, where basic descriptive data is available as CSV files.
Speaking of the visualisation the SFDP algorithm turned out to be a sufficiently intuitive representation of coauthor communities.
- **Deployment** The program was deployed as a DJANGO [6] web service on a DEBIAN7 virtual machine in the MIF CLOUD. This is further described in Section 5.

3 EXPLORATION

In order to have a better impression of the DBLP database and be able to validate the XML parser I explored the data and created basic descriptive statistics[1] with PYTHON's NUMPY package.

Table 1
Authors and Publications

Mean publications per author	5.1073
Variance of publications per author	220.7734
Median of publications	1.0
Mode of publications	1

Table 1 describes the relation between the authors and the number of their publications, while Table 2 deals with the number of the coauthors.

Table 2
Authors and Coauthors

Mean of coauthors	7.9322
Variance of coauthors	280.8594
Median of coauthors	4.0
Mode of coauthors	2

4 VISUALISATION OF COAUTHORSHIP GRAPHS

The default DBLP interface lists the coauthors alphabetically in a *"Coauthor Index"*. Additionally a colored "group indicator" is provided based on a heuristic defined as follows:

Definition 1: Consider the neighbourhood graph consisting of the target person and all (direct) coauthors of that person. Two nodes in this graph are connected by an edge if and only if they are coauthors (or co-editors) of a publication listed in dblp. Now remove the node of the target person and all incident

[1] Data downloaded on May 28, 2014

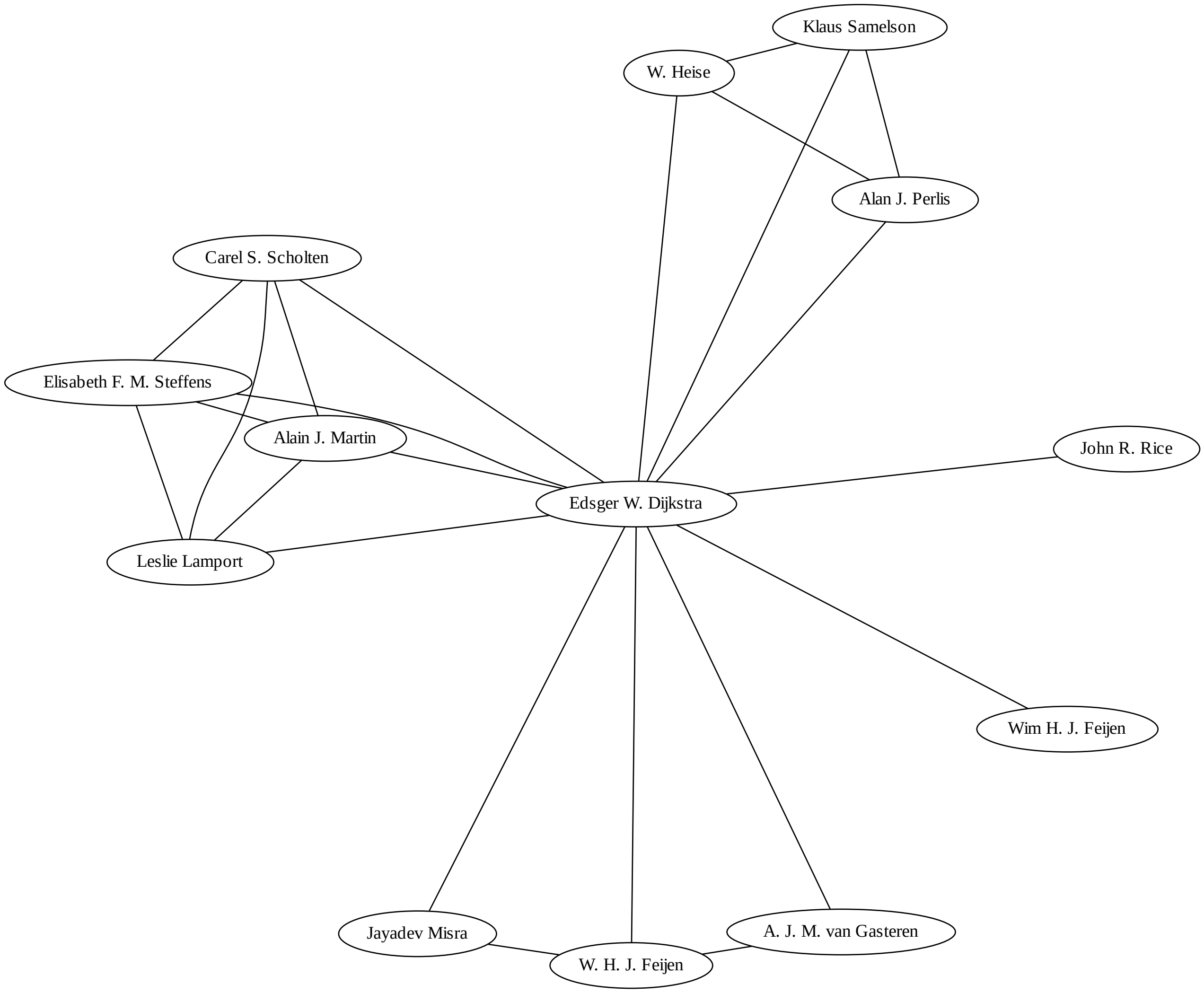

Figure 1. Coauthors of Edsger W. Dijkstra

edges. Each component of the remaining graph is identified as a community. [7]

Definition 1 is very straightforward, but it has a major drawback when it comes to visualisation. The center of the graph (the author that connected the now disconnected components) disappeared. To deal with that I found the following solution: I leave the center node in the graph, but align the connected groups circular around it. With the use of the powerful open source graph visualisation tool Graphviz [8] the components are distinguishable from the author in the middle.

A relatively small graph can be seen in Figure 1. It depicts the coauthors of *Edsger W. Dijkstra*, a winner of the TURING AWARD after whom a famous shortest path algorithm on weighted graphs is named.

Immediately one can verify that by Definition 1 there are three groups and two further single leaves of the graph.

In Figure 2 one can see the coauthors of the coauthors of *Edsger W. Dijkstra*. Note that in contrast to Figure 1 the connections between the leaves ($\hat{=}$ coauthors of coauthors) of the graph are omitted to have a lightly more clear representation. With this high number of nodes it becomes more and more impossible to visualize the graph and the perception – although it is possible to seamlessly zoom in – is auto-

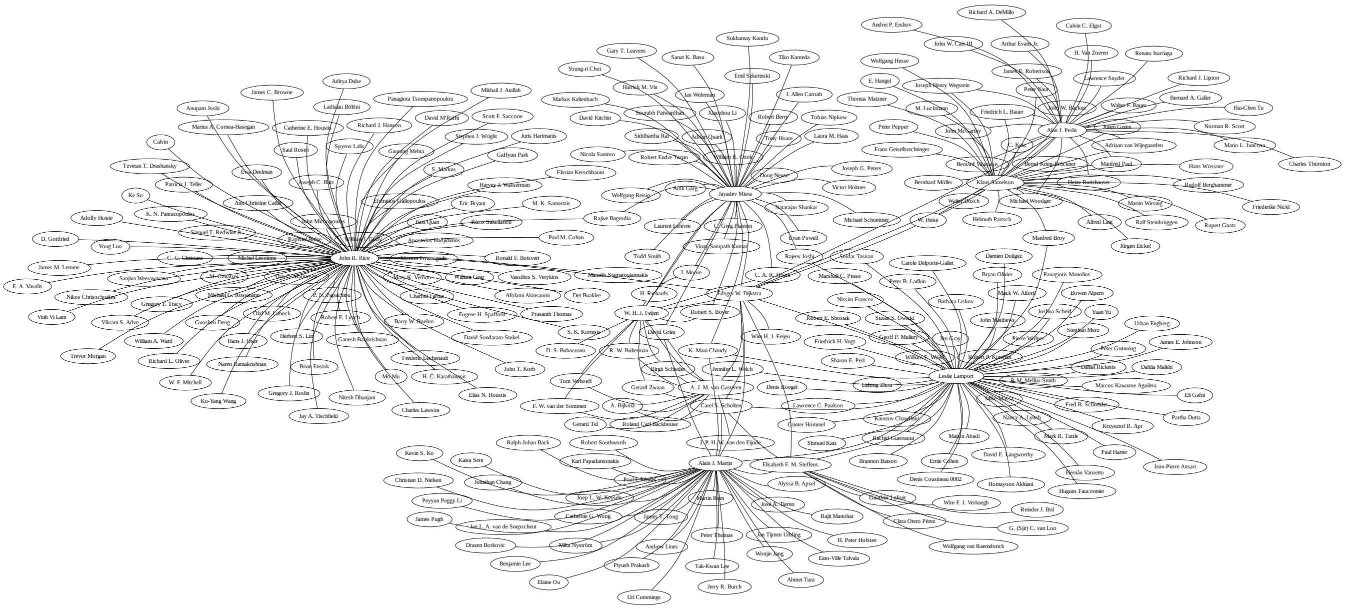

Figure 2. Coauthors of Coauthors of Edsger W. Dijkstra

matically shifted to a higher level. Instead of looking at coauthorship connections between single persons, the focus lies on the main hubs with many spokes. These hubs usually represent professors or editors of conferences.

5 Web Service

In order to have a graphical user interface for the previously described algorithms, I created a simple web service written in DJANGO [6], the web framework for the PYTHON programming language. It presents the results of the data mining process described in Section 2 and enables the user to make queries for authors of the database.

The search results show all coauthors of the query together with the number of mutual publications. Also the coauthorship graph is shown and can be downloaded together with the coauthors of coauthors graph in the PDF format.

A live demo can be seen online at http://dblpgraphs.lynyus.org.

6 Conclusions & Future Work

Parsing such a huge database requires some smart design decisions. The most important principle is to throw away all unneeded data sets as early as possible and use highly optimized libraries. The PYTHON SAX parser is a good example for this, although UNICODE support is still an issue. Converting the strings to UTF-8 nearly doubled the parsing time.

This work had a narrow focus and could be expanded in various ways: The visalisation could take the intensity of collaboration into account by coloring the edges based on the number of mutual publications. Another interesting analysis would be to have a look in which journals authors publish often.

Finally the web service should be considered as a prototype. Some improvements should be done when it comes to deployment and perfomance. A combination of an APACHE web server and a Varnish caching server seems to be a reasonable choice. Also it might be worth thinking about moving the generation of the graphs to the user using client-side GRAPHVIZ libraries like CANVIZ [9].

References

[1] M. Ley, "DBLP - some lessons learned," *PVLDB*, vol. 2, no. 2, pp. 1493–1500, 2009.

[2] H. Bast, M. Celikik, and I. Baumgarten. (2014, May) Completesearch dblp. [Online]. Available: http://www.dblp.org/search

[3] J. Diederich, W.-T. Balke, and U. Thaden. (2014, May) Faceteddblp. [Online]. Available: http://dblp.l3s.de/dblp++.php

[4] M. Georgescu, D. D. Pham, C. Firan, and E. Diaz-Aviles. (2014, May) Free search. [Online]. Available: http://dblp.kbs.uni-hannover.de/dblp

[5] R. Wirth, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000, pp. 29–39.

[6] Django Software Foundation. (2014, May) Django – the web framework for perfectionists with deadlines. [Online]. Available: http://www.djangoproject.com

[7] DBLP. (2014, May) How does DBLP detect coauthor communities? [Online]. Available: http://dblp.uni-trier.de/faq/How+does+dblp+detect+coauthor+communities.html

[8] A. Bilgin, J. Ellson, E. Gansner, and Y. Hu. (2014, May) Graphviz. [Online]. Available: http://www.graphviz.org/

[9] R. Schmidt. (2014, May) Canviz – javascript library for drawing graphviz graphs to a web browser canvas. [Online]. Available: https://code.google.com/p/canviz/