Improving Job Search by Network of Professions and Companies[⋆]

György Frivolt and Mária Bieliková

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
{frivolt,bielik}@fiit.stuba.sk

Abstract. This paper describes an approach for improving job search by generation and visualization of networks of companies and professions. Both networks are produced from job offers, which always contain information regarding the required profession and the company interested in an employee. The network reflects discovered relations between professions and companies by analyzing common professions of job offered by different companies. Visualization of discovered relations helps the user in better navigation and therefore better user experience when searching for a job.

1 Introduction

Introduction of the Web was soon followed by unforeseen expansion of publishing of various information types. Nowadays the Web provides a colorful information and knowledge market. Natural demand for searching and navigation appeared hand in hand by the dramatic increase of information sources.

We are concerned on job offers. Certain companies or agencies are interested in specific types of positions. Perhaps a software company or an agency hiring software experts is interested in positions like coder, administrator, team leader or project manager and probably does not announce job offers typical for agriculture. Differences in company profiles are showed up in the job offers they provide. We propose an approach to discover relations between professions and companies by analyzing the network of professions and companies. The nodes in a network of professions are connected providing job offers of particular company that contains both professions. The aim of this paper is present an approach for producing such networks of professions (jobs, positions) and companies.

Splitting the network of professions and companies into clusters and subclusters is crucial in the navigation of the data set of job offers. Users usually appreciate intuitively categorized information providing few, but clear choices.

[⋆] This work was supported by Science and Technology Assistance Agency under the contract No. APVT-20-007104 and State programme of research and development "Establishing of Information Society".

We follow the idea that identification of professions and companies similar to each other would help the user to find the job offer what he is interested in. Clustering can be proceeded automatically without any human intervention.

This work is carried out within a larger project aimed at developing a set of software tools that process information in a heterogeneous environment such as the current and the future Web [3]. The set of software tools covers the whole process of data gathering, analyzing, organizing and presenting in particular application domain [9]. Justification of the results is being provided on the domain of labour market. The intention is to provide information and knowledge on job offers.

The tool described in this paper aims to provide an integrated view of job offers, which are visualized as network of companies and job positions. There are several issues to be touched:

- wrapping and integrating job offers from web sites,
- creation of graph of companies and job position, respectively,
- visualization of the graph.

The paper is structured as follows. In Section 2 we discuss sources for this work. Next, Section 3 discusses the the processing the initial data from web sources in order to find interesting relations through visualization of networks of companies and professions. The principle of network generation is described in Section 3.2. The relations are generated from an ontology developed for the job offers domain and from web pages, Section 3.1 discusses this issue. Clustering is described in Section 3.3. Finally we give conclusions and plans for future works.

2 Related works

We mention different tools which work either in the domain of job offers and offer interesting ideas related to network structure of the users, or integrate various offers from more sources. As improvement of navigation within clusters is naturally provided by visualization, efforts for visualization of data stored in a networks are also mentioned.

Project TouchGraph aims to visualize different networks either in a Java applet or in a stand alone application [14]. Several applications are maintained by the project. It is possible to visualize Wikipedia topics (*www.wikipedia.org*), here the vertices are Wikipedia topics and the edges are references pointing to further topics. Other application of TouchGraph is visualization of web pages' network. This network is gathered from the Google cache. Pages crawled by Google are cached and this cache is accessible.

LinkedIn provides a sophisticated search over social relations for the purpose of finding the right contact persons to get a job [7]. The users build their network by giving people related to them (friends, old colleges, family relatives). Job search consists of an identification of possible employers and chains getting the user connected with the user. After finding the chain, the user is supposed to contact the employer through the chain of friends. This service operates on the social network those who seek and those who offer jobs.

One of the first inspiration of our work was the project *www.buzztracker.org.* The project aims at wrapping of news and their consequent visualization [8]. The Buzztracker system gathers news from the Google world news directory and visualizes frequencies and relationships between locations. A map of locations is generated from the rough news data. A location receives higher weight if more news appear in this location. Two locations get connected by a weighted edge if the location appears in the news of the other location. Visualization of the news maps shows the sense of useful presentations. Showing the world's recent "buzz centers" helps the user to navigate in huge amount of news. We believe that better navigation can be yielded on the field of job offers as well as for the news by relying on the knowledge gathered from a topology of profession and company relations. However, the task of visualization of news network differs from the visualization of job positions and companies. We are not mapping the vertices to a map. The vertices do not have their predefined coordinates, like cities on the world map in case of buzztracker.

Considering millions of job offers, we obtain a massive network that should be analyzed. Several tools implementing algorithms for network analysis and visualization are being developed (Pajek [1], JUNG [12], InFlow [6], Cyram NetMiner [13]). These tools mostly provide functions such as ranking vertices according to their importance, centrality measurements, clustering and visualization of the network and other functionalities.

3 Tool for job clusters discovering

We have proposed and developed a software tool that based on network of job offers discovers interesting relationships between them and in order to find interesting clusters of nodes representing companies or professions. A cluster represents the set of nodes that are more interconnected among each other than between other nodes in the network.

Figure 1 illustrates the processing of the initial data from web sources containing job offers. The output of the processing is a visualization of the network of professions and companies.

The processing consists of the following steps:

- *Acquiring job offers.* Job offers are stored in an ontology which has structure covering the most attributes relevant for the job offer domain. The ontology can be accessed by parsing the OWL files (e.g., using Sesame framework for querying and inferring). At the moment the ontology is filled primarily manually. We have implemented also a wrapper for existing job portal. In this case the job offers available on web pages are wrapped to the XML form and processed further. See Section 3.1.
- *Network generation.* Job offers are used for forming a bipartite graph with two types of vertices: professions and companies. Section 3.2 discusses how the networks of professions are generated from the bipartite graph.

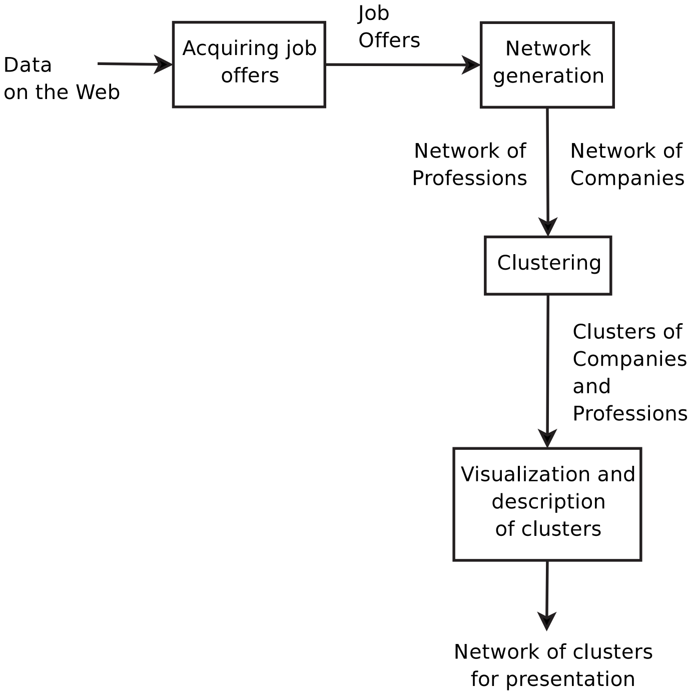
Data
on the Web
Acquiring job offers
Job
Offers
Network generation
Network of Professions
Network of Companies
Clustering
Clusters of Companies and Professions
Visualization and description of clusters
Network of clusters for presentation

Fig. 1. Process of job clusters discovering.

- *Clustering.* The vertices of the networks are split into clusters, i.e., groups of vertices that are more interconnected among each other than with the rest of the network. Section 3.3 discusses our clustering approach.
- *Visualization and description of the clusters.* The results, mined clusters of companies and job positions are supposed to be presented to the user. The map rendered towards the user is supposed to be clickable. The output has to be therefore in a picture and in an imap format, which is processed on the server side in our case.

The tool as such is designed as generic, i.e. it can process various networks. Individual network types influence the step of gathering relevant documents from the web sources together with wrappers.

The tool is implemented in *Python*, the graphs are represented using free library *pygraphlib*. Retrieving the source from which the graph is generated latter is solved by *Redland Python RDF - librdf* module. As in the future the data will be retrieved using the Sesame framework, librdf will replaced by pySesame module interfacing Sesame. The presentation of the results on the web is solved using *mod_python* apache module. Rendering the graph to picture and the imap format is currently done by the *graphviz* library, but perhaps a more interactive solution will be chosen in the future such as *TouchGraph* [14].

3.1 Acquiring of job offers

Job offers are stored in an ontology, which was created using the *Protégé* ontology editor. Our tool relies on the data stored in the ontology. Among several defined classes we are the most concerned about classes `jo:Organization` and `jo:JobOffer`. Class `jo:Organization` describes the attributes of the organization (company) offering the job, we take advantage of the following attributes:

- `jo:name` is the name of the organization,
- `jo:offers` are the job offers provided by the company.

Class `jo:JobOffer` contains the following attributes relevant for our purposes:

- `jo:isOfferedBy` represents the organization offering the job,
- `jo:offersPosition` represents the position offered by the organization,
- `jo:hasRequirement` represents the instance of type `jo:Requirement` defining required skills,
- `jo:hasResponsibility` represents the instance of type `jo:Responsibility` defining the responsibility.

Class `jo:Reuquirement` and class `jo:Responsibility` are both subclasses of `jo:JobOfferAttribute`.

Our tool relies on retrieving data from the ontology, which is parsed from an OWL file. As our job offers ontology is at the moment filled manually, we have implemented a wrapper for job offer web site *www.profesia.sk* to retrieve data for testing purposes. If the job offers stored in the ontology are not numerous or diverse (in sense of types of industries) enough we test our tool by retrieving the data directly from web sources using our wrapper.

As illustrate on Fig. 2 the web pages have to be first cleaned and converted to a regular XML/XHTML code, which is further queried exploiting its tree like structure and using regular expressions. The result of the wrapping is the job offer represented in XML. We use an open source project *HTMLtidy* for converting the HTML into a correct XHTML code. The XHTML code is currently parsed using *Amara* library (Amara is developed on the top on *4Suite* python library).

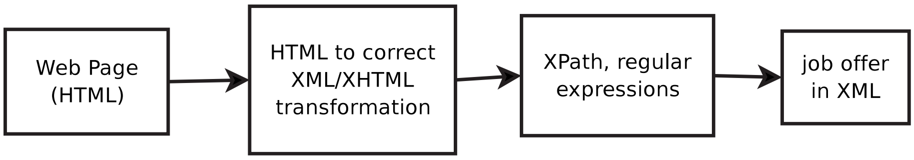
Web Page (HTML)
HTML to correct XML/XHTML transformation
XPath, regular expressions
job offer in XML

Fig. 2. Schema of the HTML page processing.

3.2 Producing the network

Job offers are currently retrieved from an ontology of job offers developed within the scope of our project [3]. The ontology includes all relevant information what an employee might be concerned about. It is represented in OWL. We are concerned about the company, job position and also possibly required skills and company industry area. Our tool for job clusters discovering requires enough data to operate on (more hundreds of job offers) and sufficient diversity of the attributes of job offers. Companies offering similar positions with the same profession requirement will result in only one cluster of positions or companies being found.

The networks of professions and companies are produced from job offers. Naturally every job offer contains the identification of a company or an agency, which seeks applicants, and the title of the job or the position which it is interested about. A bipartite graph of companies/agencies and job positions can be constructed with job offers acting as edges. A company vertex is connected with a job position vertex when they appear at the same job offer. Such a network is depicted on Fig. 3.

The network of positions (or jobs) $G'(V', E')$ is produced from the bipartite graph $G(V_{\mathrm{jobs} \cup \mathrm{companies}}, E)$ in an intuitively way:

$$V' = V_{\mathrm{jobs}},$$

$$\exists w \in V_{\mathrm{companies}} : \{u, w\}, \{v, w\} \in G(E) \Rightarrow \{u, v\} \in E'$$

where vertices u and v can be the members of V_{jobs} only.

Naturally more of vertices with the property of w could be found in $V_{\mathrm{companies}}$. The produced edges is weighted as the function of number of such vertices found. The decision what kind of weighting (if any) is optimal to use depends on the application domain. Network of companies is defined in analogical way as the network of positions. For this network the vertices, which are companies, are connected if they both offer a job with the same position.

Behalf the creation of graph from the network of companies and job positions other possibilities may be also feasible for improving the presentation towards the user. The ontology where we retrieve the job offers from contains a wide range of information describing the job offer. Attributes such as expected skills and industry (the area or areas where the company offering job act) are also candidates for being vertices of the network behalf company vertices.

The described network is supposed to contain up to few thousand vertices. For this size effective algorithms for clustering exist. For cases when application of clustering algorithms would not be feasible we developed a simple approach for cutting a part of the network [4]. The algorithm is based on a modified breath-first search. We operate on a part of the network when the chosen clustering algorithm does not scale with the whole network.

3.3 Clustering

We implemented the algorithm for clustering proposed by Newman-Girvan for retrieving the communities presented in the network of professions and companies [10]. This approach is based on removal of edges with high betweenness centrality values. A fast algorithm for computation of betweenness centrality (computing in $O(nm + n^2 \log n)$ time) was proposed by Brandes [2]. Other approach also feasible is application of hierarchical clustering methods [5,11]. The difference between these approaches and edge-removal technique is the bottom-up direction of identification of communities in the former – small communities are joined to create bigger ones – and top-down processing of the edge-removal approaches – communities, starting with the whole graph are being divided to smaller parts.

Both edge removal methods and hierarchical clustering can produce hierarchy of clusters. We need the hierarchy of clusters for:

1. rendering the cluster where a node belongs to and
2. improving navigation in the hierarchy of the clusters

A straightforward way how to represent clusters hierarchies is to define a tree of clusters behalf of the graph of positions or companies. The vertices of the graph are mapped to the leaves of this tree. Our idea of representation is not to separate the two structures (the graph and the hierarchy of clusters), but to define the cluster hierarchy in the graph. In this enriched graph the vertices will be either called clusters (or cluster vertices) or atomic vertices representing job positions or companies. Hence four types of edges can be defined on this structure:

1. the original undirected edges of the graph connecting atomic vertices,
2. directed edges of the cluster hierarchy connecting clusters,
3. directed edges between the leaves of the cluster hierarchy and the atomic vertices of the original graph – these edges define the belonging of the vertex to the cluster,
4. undirected edges between the vertices representing clusters on the same level, i.e. same depth in the cluster hierarchy – these weighted edges describe the proportion of the edges between the clusters.

We do not consider the last type of edges (edges between clusters) in the visualization. The cluster hierarchy is defined by directed edges with vertex in higher depth (representing narrower cluster) in its head. The original vertices representing positions or companies are connected by undirected edges. Every vertex except of the root cluster vertex of the cluster hierarchy has exactly one incoming directed edge, the root cluster vertex has not any incoming edge.

Vertex is rendered based on its type. If the vertex represents a cluster, its subclusters or subatomic vertices (to both directed edges leads from the cluster vertex) are rendered. If the vertex is an atomic vertex than the only cluster vertex, from which the atomic vertex gets the only incoming edge, is rendered. Described approach resolves both requirements for clustering usage as described above in this subsection.

4 Summary

An approach to improve navigation in job offers is described in this paper. We developed software tool that realizes proposed approach. The tool wraps job offers from public web sources, generates network of professions and companies, and finally generates a hierarchy of professions and company profiles, respectively. This hierarchy can be either listed in textual form or visualized as a map.

Described tool should help the user to find job offers more effectively. The tool automatically identifies clusters of professions and companies. By the clustering the job offers are categorized based on the professions and companies, which helps the user in better navigation. Visualization of the network is a further added value to the simple categorization of the job offers.

A strategy of setting the weights of the edges between the professions and companies will be developed. Also a design and implementation of a simple visualization of the generated networks is necessary to develop. The network presentation should be aware of the revealed clusters. Vertices belonging to the same cluster should be distinctively close to other vertices in the same cluster.

References

1. Vladimir Batagelj and Andrej Mrvar. Pajek, February 2005. Available at http://vlado.fmf.uni-lj.si/pub/networks/pajek.
2. Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.
3. Pavol Návrat et al. Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources. Technical report, Slovak University of Technology in Bratislava, June 2005. State programme of research and development "Establishing of Information Society", report-phase2.
4. Gyorgy Frivolt and Mária Bieliková. A community-cutting approach. In V. Svatek and V. Snasel, editors, *RAWS 2005 - Proc. of the 1st Int. Workshop on Representation and Analysis of Web Space*, pages 49–54, September 2005.
5. Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *Physical Review Letters*, December 2001. cond-mat/026113.
6. Valdis Krebs. InFlow – Java Universal Network/Graph Framework, November 2005. Available at http://www.orgnet.com/.
7. LinkedIn, November 2005. Available at https://www.linkedin.com/home?trk=logo.
8. Craig Mod. Buzztracker: Word news, mapped, August 2005. Available at http://www.buzztracker.org.
9. Pavol Návrat, Mária Bieliková, and Viera Rozinajová. Methods and tools for acquiring and presenting information and knowledge in the web.
10. Mark E. J. Newman. Detecting community structure in networks. *The European Physical Journal B*, 38:321–330, 2004.
11. Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review Letters*, 2004. cond-mat/026113.
12. Joshua O'Madadhain, Danyel Fisher, and Tom Nelson Jens Krefeldt. JUNG – Java Universal Network/Graph Framework, November 2005. Available at http://jung.sourceforge.net/.

13. NetMiner project team. Cyram NetMiner, November 2005. Available at http://www.netminer.com/NetMiner/home_01.jsp.
14. Alexander Shapiro. TouchGraph, November 2005. http://www.touchgraph.com/.

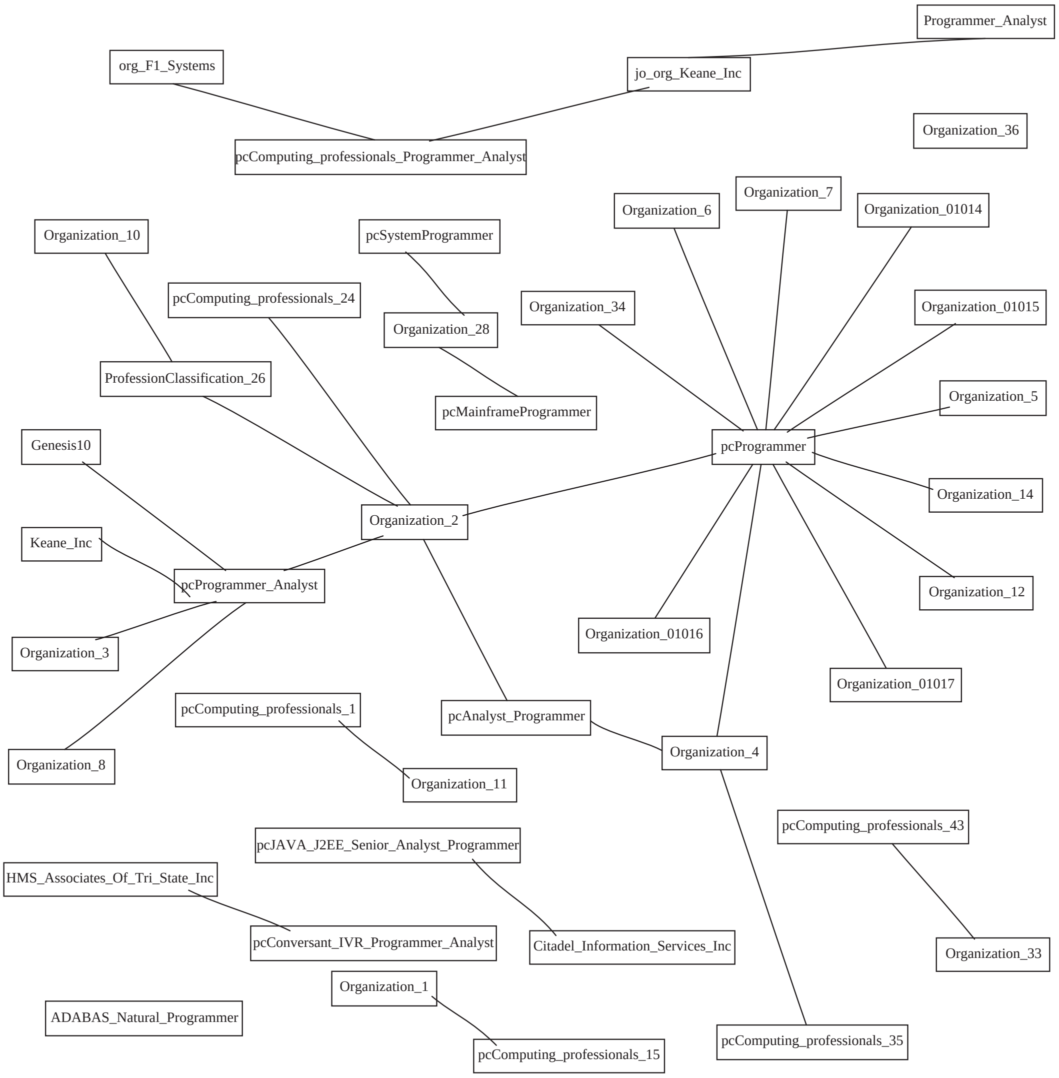

Fig. 3. An example of bipartite graph of companies and job positions.