
# A2-D9

# Job search with GoPubMed

| | |
|---|---|
| Project title: | Reasoning on the Web with Rules and Semantics |
| Project acronym: | REWERSE |
| Project number: | IST-2004-506779 |
| Project instrument: | EU FP6 Network of Excellence (NoE) |
| Project thematic priority: | Priority 2: Information Society Technologies (IST) |
| Document type: | D (deliverable) |
| Nature of document: | R (report) |
| Dissemination level: | PU (public) |
| Document number: | IST506779/Dresden/A2-D9/D/PU/b1 |
| Responsible editors: | Matthias Zschunke |
| Reviewers: | Michael Schroeder |
| Contributing participants: | Dresden and Transinsight |
| Contributing workpackages: | A2 |
| Contractual date of deliverable: | 29 Feb 2008 |
| Actual submission date: | 29 Feb 2008 |

**Abstract**
GoPubMed is a successful semantic web application in the life sciences. The goal of this deliverable is the application of the GoPubMed technology in the area of Job search, where a different ontology and document corpus are needed. We describe two approaches to job search with semantics: "Jobs for people" and "people for jobs". In the former prototype, we have built a job ontology comprising geographic information, terminology on skills and types of positions. As document corpus we have used job ads available from public web sites like Nature jobs. The second engine "people for jobs" aims to identify scientists matching a given job profile. To this end, we cluster 17.000.000 PubMed abstracts to identify several millions of author profiles. Using GoPubMed, authors are linked to their research area, co-authors are identified, the journals they pblish in and for internationally leading authors their field of leadership is listed. "People for jobs" is built into GoPubMed and freely available over the web at www.gopubmed.org.

**Keyword List**
GoPubMed, job search, ontology, textmining

*Project co-funded by the European Commission and the Swiss Federal Office for Education and Science within the Sixth Framework Programme.*

# Job search with GoPubMed

**Matthias Zschunke**[Dre,TI], **Lars Ackermann**[TI], **Michael R. Alvers**[TI], **Liliana Barrio-Alvers**[Dre,TI], **Matthias Leis**[TI], **Jan Mönnich**[TI], **Michael Schroeder**[Dre]

[Dre] Technische Universität Dresden, Germany, [TI] Transinsight GmbH, Dresden, Germany,

29 Feb 2008

**Abstract**
GoPubMed is a successful semantic web application in the life sciences. The goal of this deliverable is the application of the GoPubMed technology in the area of Job search, where a different ontology and document corpus are needed. We describe two approaches to job search with semantics: "Jobs for people" and "people for jobs". In the former prototype, we have built a job ontology comprising geographic information, terminology on skills and types of positions. As document corpus we have used job ads available from public web sites like Nature jobs. The second engine "people for jobs" aims to identify scientists matching a given job profile. To this end, we cluster 17.000.000 PubMed abstracts to identify several millions of author profiles. Using GoPubMed, authors are linked to their research area, co-authors are identified, the journals they pblish in and for internationally leading authors their field of leadership is listed. "People for jobs" is built into GoPubMed and freely available over the web at www.gopubmed.org.

**Keyword List**
GoPubMed, job search, ontology, textmining

# Contents

# Contents

# 1 Introduction

Searching a job is still a hard task. There are lots of job portals in the world wide web most of them with rather limited search functionalities. Searching for simple keywords often fails to retrieve the proper job ads. Usually searches yield no results or a rather long list of non-relevant ads.

The job search portals perform very different on the task of job search. There is a large variety of machines with different features ranging from simple search sites to portals where people can register and get announced if new appropriate job ads are published. We analyzed job search websites and provide some drawbacks and limitations on current job search.

Let's start with a common job search site, the site of the German job center, *Arbeitsagentur*. The job searcher can search for the job type and the profession or education. There is the possibility to restrict the results by time period and location radius. There is no keyword search in job descriptions which is a large drawback. The job searcher needs to commit oneself to a profession name and cannot find jobs with regard to its qualifications. It is not possible to select multiple sectors or professions at a time. The search engine does not provide jobs with regard to the qualification *profile* of the job searcher.

Another common career portal is Monster (www.monster.com) which gives the job searcher the possibility to create and publish a profile of itself. This includes the curriculum vitae as well as terms that describe the personal education and knowledge. The profile can be used by job posters to find persons that may be good candidates for the offered job.

*Nature Jobs* for example is specialized on job ads in the fields of research and scientific professions. Searches can be performed by keywords on the different attributes of job ad, e.g. on the title, description the employer or the location. *Nature Jobs* provides also a way to search for tags. These are specialized keywords that describe the job ad. The tags are provided by the job posters and the *Nature Jobs* staff as well. A drawback here is the fact that tags are not reviewed or controlled, e.g. both tags *Alzheimers* and *Alzheimers disease* are provided in *Nature Jobs* both with a different listing of job ads. For both tags there are 2 job ads each and neither of them is listed also for the other tag. That means searching for *alzheimers* will not find the same job ads as the search for *alzheimers disease* although both denote the same issue. Another example: a search for *postdoc* yields not the same job ads as a search for *postdoctoral* (as in *postdoctoral position*). The keywords are always matched completely, there is no stemming of words (see *leading position* vs *leader*). *Nature Jobs* offers the possibility to select the tags directly from a list and get the corresponding jobs, but this procedure is inconsistent with the results obtained with the advanced search typing the tag into the *tags field*. A problem here is that it matters in which input field you type the term, it makes a difference to search for *alzheimers* in description or in the list of tags. The tags are not automatically assigned to job ads.

Currently the job search engines do not provide the possibility to filter the job ad by keywords also regarding synonymous mentionings or words with similar meaning. To address the problems of current job search machines it would be preferrable to utilize controlled vocabularies to the job search, that means a system that uses semantic connections between the search terms to reflect synonyms and conceptualization of the search terms.

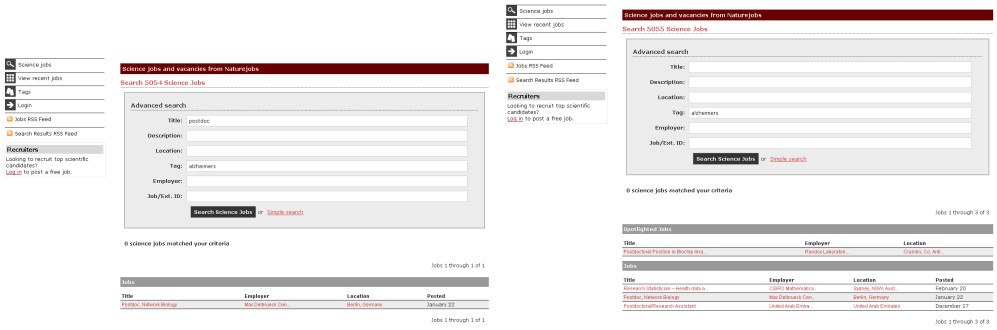


Figure 1: Limitation of job search in *Nature Jobs*. Left: Search for *postdoc* in the job titles and *alzheimers* in the tags. Right: Search for *alzheimers* without *postdoc* also showing *postdoctoral positions* that are not found by the query *postdoc*.

## 2 Creating the job ontology and jobs

To address the problems with current job search we created a job search engine that utilizes job ad specific background knowledge to filter the job ads by search terms. The knowledge is represented in form of an ontolgy containing over 8000 terms that are hierarchically structured.

As a test system we extracted about 700 job ads from the career portal of www.sciencemag.org. These job ads are used to create the ontology. We extracted term candidates from the job descriptions and used them to setup an ontology for the topic of job search. This task was performed with an ontology editor integrated into the job search application. This editor enables the user to insert, delete, and edit the terms and the conceptual structure of the background knowledge. See figure 2.

Job ads usually consist of a title and a description. The title ususally names the position. The description gives information about the company, the position objective, the location and lists the technical requirments and soft skills of the desired candidate.

The ontology consists of common job search concepts such as skills, job types, activities, locations as well as specialized concepts for the domain of scientific work. The latter concepts are taken from the MeSH headings. We did not use the complete MeSH Ontology but extracted some branches that are relevent in scientific job ads. In the category *skills* you find *soft skills*, *technical skills*, *education*al concepts amongst others. *Job type* refers to *full* and *part time positions*, *apprenticeship*, *internship*, etc.

The goal with our approach is to filter the available jobs by the terms given in the background knowledge taking parent-child relations and synonyms into account. It is possible to filter jobs by position types like postdoc or leadership as well as keywords describing the tasks and other job ad related information.


Figure 2: The ontology editor. Terms can be inserted, deleted and edited. Synonyms can be defined and the hierarchical structure can be changed simply with drag and drop actions.

There are still open problems. Although we apply sophisticated text-mining algorithms mapping the job ads to the ontological background knowledge there are terms that are still

ambiguous in their meanings. For example, the term *apprenticeship* can determine the *job type* as well as a *requirment* to the candidate to get the job.

# 3 Use cases

In job search there are a couple of concepts to consider. To search for a keyword or the job title will either yield a long list for jobs being not relevant or it will say "no results". The machine needs a kind of intelligence to decide which jobs are relevant for you. For example if you are looking for a leading position you need to consider all keywords denoting the leadership, such as *lead*, *team leader*, *group leader*, *team management*, *project management*, *manager*, *director* etc. Or maybe you are about to finish your PhD and thus you are looking for a *post-doctoral fellowship* or a *postdoc position*, both terms refer to the same concept.

Maybe you are looking for a job in the fields of *neoplasms*. Searching with a usual job search engine will not find all jobs on *neoplasms* since there are synonyms (e.g. *cancer*) and lots of sub-concepts that could be mentioned in the job ad, e.g. *acute leukemia*, *prostate cancer* or *breast cancer*.

The ontology here can help to automatically solve the task of resolving these conceptual overlaps. An ontology can here be used to sort the articles by relevance to given terms. You may select postdoc and you will get all job ads mentioning *post-doctoral position* and related terms. Or you select *Neoplasms* and will retrieve all job ads containing terms as *cancer*,

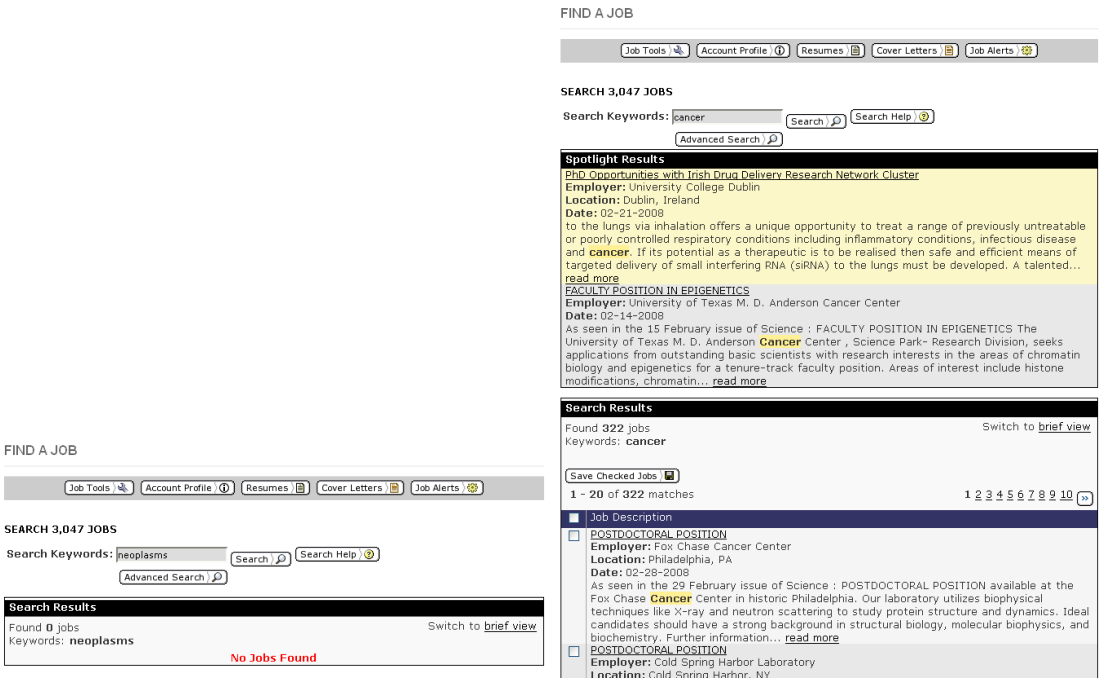


Figure 3: Problems in job search. A search for *neoplasm* retrieves nothing whereas the synonymous keyword cancer could be found in over 300 job ads.

# 4 The JobJob prototype

Our job search machine uses the previously defined structured vocabulary to filter the 721 job ads by job-relevant terms. See figure 4 for an overview of the ontology as well as the list of job ads.

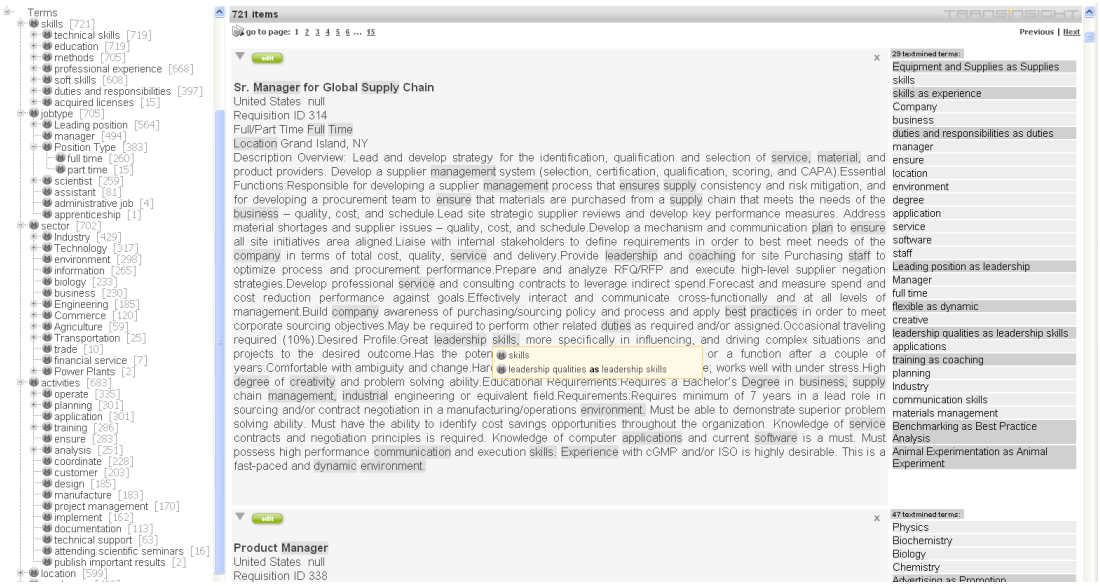


Figure 4: Our job search machine. Left: the hierarchical view of the job specific background knowledge, extended to the first 2 levels. Right: the view ot the list of job ads, found ontology terms are hightlighted, the tooltip shows the annotation by our machine: here *leadership qualities* could be found in the term *leadership skills*.

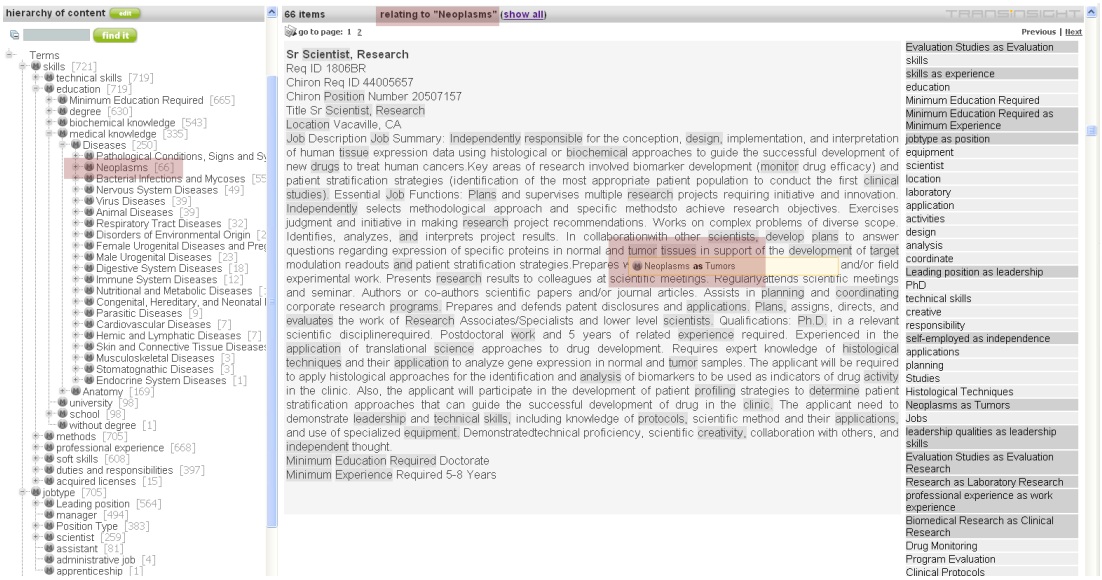


Figure 5: Filtering by *neoplasms* also finds job ads where *tumor* is mentioned in the description text.

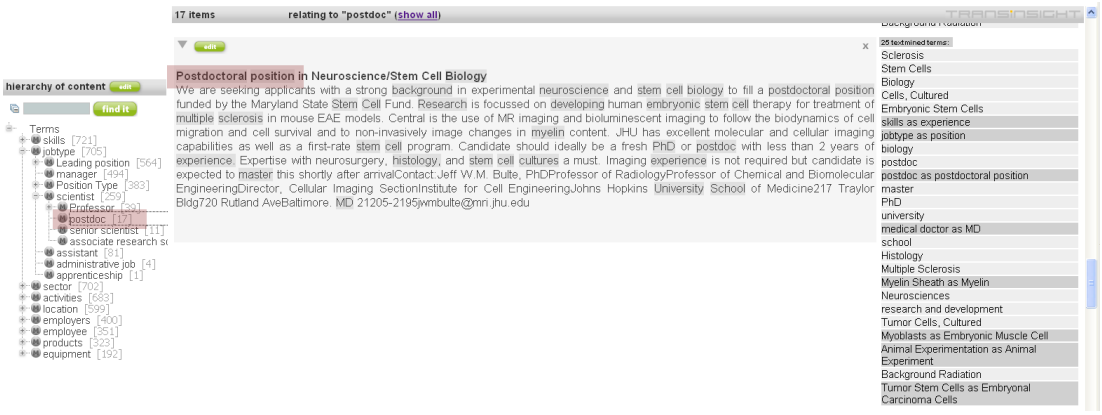


Figure 6: The ontology-based job search machine. Treeview on left side: the hierarchical view of the background knowledge, in square brackets the number of job ads matching the selected term. On the right side: the job ads with all matching terms from the background knowledge highlighted. The term *postdoc* is found synonymously in the word *post-doctoral position*.

# 5 Author profiles: Finding People for Jobs

## 5.1 Motivation

From the perspective of a job searching specialist the large disadvantage of the job portals are the rather limited search functionalities, which often actually are restricted to a search of job titles. From the view of the employer searching for a competent specialist in a field it is a hard task to filter the profiles of job search people in the job portals. Maybe the right specialist for a given job is not registered in the portal you scanned or is not searching a job right now. So one can think of an appliation that finds the right person for a given job profile.

Finding the right person for a job is quite simple if you know everybody that could be a candidate. Just look at the profiles and select the person that best fit your needs. But usually you do not have any such information.

You can try to find someone with one of the job portals such as monster. But you also know that not everybody that might be a good canditate has registered in such a portal and there is a large quantity of portals.

For the fields of research jobs in biomedical sciences there is a source where we can find all people that might be good candidates for a given job. We are looking for scientists and therefore we can use the information provided by the papers that are written by them. For the fields of biomedical research there is PubMed as a rather complete resource of citations to scientific biomedical literature. The articles listed in PubMed give deep information about the people that have written them.

The idea is to match the job profiles (i.e. extracted terms denoting *skills*, *experience*, *location* and *job type*) against the author profiles that contain information on the topics, the location the author is affiliated with, his status (i.e if it is a senior author performing leading research in certain topics or if it is a PhD student or a postdoc with only few articles). Depending on the number of matching concepts it would be possible to create a match-score for job vs. author profiles.

The information necessary to do the task must be extracted from the articles listed in PubMed. The big problem here is to catch the authors. PubMed only lists author names. And these are very ambiguous, e.g. the name "R. Smith" appears in more than 10 thousand articles.

## 5.2 Approach

We applied an disambiguation algorithm for authors to the articles given in PubMed. The similarity measure is based on article and author name features, e.g. coauthor names, journal names, keywords (MeSH headings), words from the paper title and the affiliation string.

The algorithm [Torvik et al., 2003] computes the probability that a pair of articles is authored by the same person which depends on the number of features the articles have in common. The more common two articles with a given author name, the higher the probability that the name denotes the same person. For any given author name we compute the pairwise similarity values for the corresponding set of documents. These are clustered subsequently by two clustering approaches: first we guess a threshold of the similarity values and only select pairs of articles that have a higher value. Since there is a chance that two articles have a high similarity although there are two distinct authors we subsequently applied a simple Markov clustering approach.

Once the disambiguation step is performed the articles written by an author can be taken to retrieve information about the author. For each we can compute a profile that describes not

just the authors work but also where the author worked. The list of major affiliations with temporal range is extracted as well as the keywords for which the author is internationally leading, if so. For an author it also possible to guess his status: An author with many articles on which he even appears as a last author a couple of times can be said to be a senior. So only group leading positions should be offered. On the other hand an author with only few article who exclusively appears as a first author (or at one of the first places on the co-author list, respectively) is maybe a PhD student or a postdoc looking for positions as postdoc or research assistant.

## 5.3 Results

In PubMed we found over 3 million different names related to over 17 million articles. About 40 names are associated to over 10 thousand articles. There are about 3600 names associated to more than 1,000 articles. Over 2 million names are subject to disambiguation since these are associated to 2 or more articles. We computed more than 15 million authors. About 1 half of the names are each clustered in a single group which means that there is only one person with that name. There are actually about 300 authors that have written more than 1,000 articles.

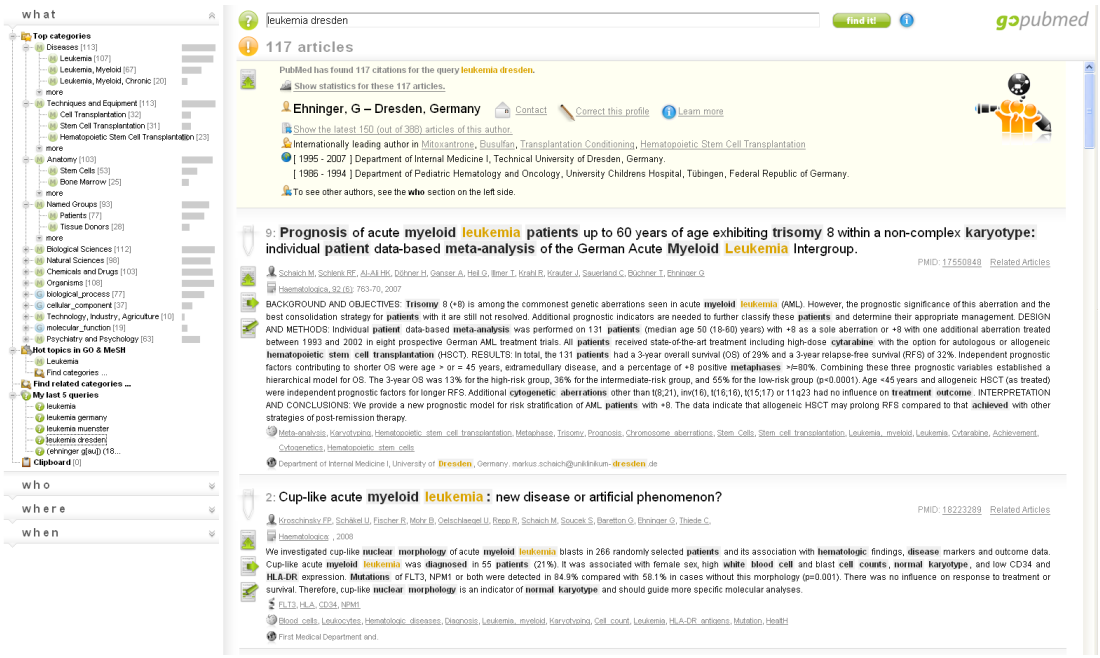


Figure 7: Author profile extracted from PubMed articles. The top author for leukemia in Dresden.

## 5.4 Edit Profiles

In order to improve the profiles extracted from the PubMed abstracts we provide the users with the possibility to manually edit the automatically created profiles. In cases where the disam-

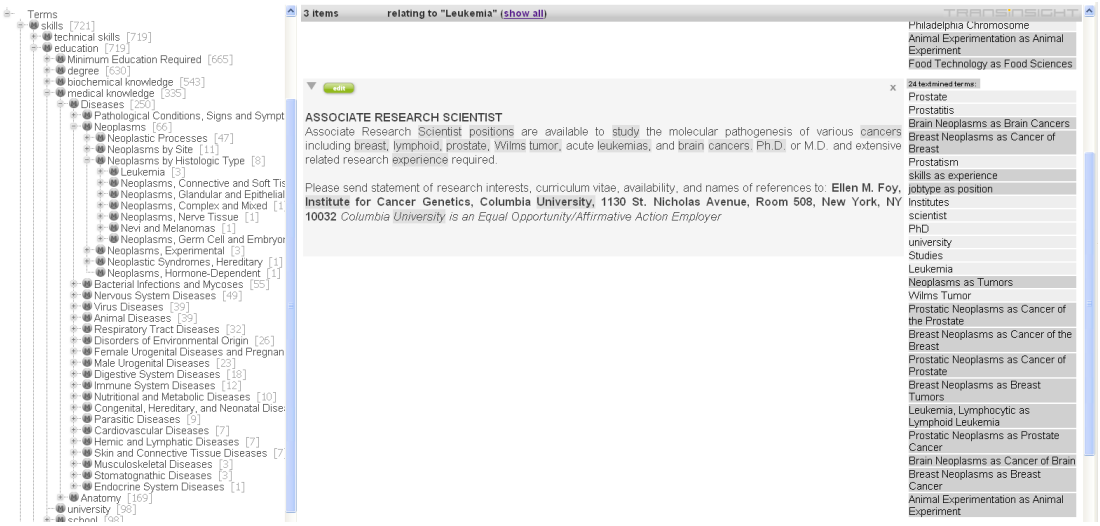


Figure 8: Job profile with highlighted concepts from the job ontology. Here the focus lies especially on terms that are also present in MeSH, e.g. *leukemia* and *brain neoplasm/cancer*. On the left the part of the concept hierarchy is shown containing the term *leukemia* that was used here to filter the 720 job ads. On the right one of the three remaining the job ads is shown.

biguation algorithm or the information extraction procedures do not perform on a precission of 100% the users are encouraged to refine the information given in the profiles as well as the assignment of PubMed articles to specific authors. So it is possible to give a title, adjust the first and last name, provide an email address and to complete and refine the list of affilitions. Furthermore, by a single click on the checkbox icon in front of each article the user can decide whether an article was coauthored by the author or not. See figure 9.

# 6 Conclusion

GoPubMed is a flexible, general system ,wh ich can be adapted for other domains. This deliverable has illustrated the application of the GoPubMed technology in job search. In the first prototype, a job ontology was built using the ontoogy editor. The ontology comrprises some 8000 terms. The terms were mapped against job ads from the Science/Nature job search. A second prototype was build to identify people suitable for jobs. The core of this engine are author profiles for millions of authors. The profiles have been automatically dereived from 17.000.000 abstracts. Author profiles are included in GoPubMed and freely available via www.gopubmed.org.

# References

[Torvik et al., 2003] Torvik, V., Weeber, M., Swanson, D., and Smalheiser, N. (2003). A probabilistic similarity metric for medline records: a model for author name disambiguation. *AMIA Annu Symp Proc*, page 1033.

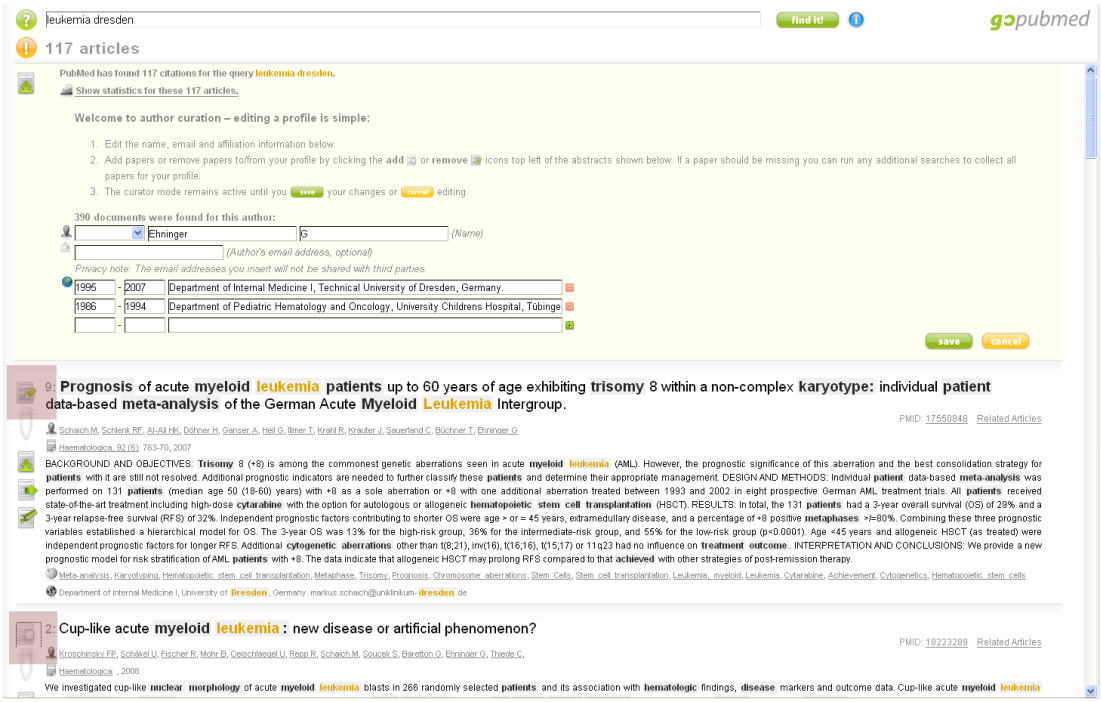


Figure 9: Edit author profiles. If an extracted profile is not perfectly correct it can be improved manually. The icons used to add/remove articles to/from authors are marked with red squares.