

ArnetMiner: An Expertise Oriented Search System for Web Community

Jie Tang, Jing Zhang, Duo Zhang, Limin Yao, Chunlin Zhu, and Juanzi Li

Department of Computer and Technology, Tsinghua University
{tangjie, zhangjing, zhangduo, ylm, ljz}@keg.cs.tsinghua.edu.cn

Abstract. Expertise Oriented Search aims at providing comprehensive analysis and mining for people from distributed sources. In this paper, we give an overview of the expertise oriented search system (ArnetMiner). The system addresses several key research issues in extraction and mining of a researcher social network. The system is in operation on the internet for more than one year and receives accesses from about 1,500 users per month. Feedbacks from users and system logs indicate that users consider the system can really help people to find and share information in the web community.

1. Introduction

Web-based communities have become one of the most important online applications [3] [5]. Web community targets at providing user-centered services to facilitate finding and sharing information. Previous information search and mining methods is not sufficient in this new scenario, due to lacks of semantics and lacks of effective and efficient approaches to deal with the new mining issues.

In this paper, we present a novel expertise oriented search system for web community, which is available at <http://www.arnetminer.org> [7]. Our objective in this system is to provide services for searching and mining the semantic-based web community. Specifically, we currently focus on academic researcher community and aim at answering four questions: 1) how to automatically extract the researcher profile from the existing unstructured Web, 2) how to integrate the information (i.e., researchers' profiles and publications) from different sources, 3) how to provide useful search services based on the constructed web community, and 4) how to mine the web community so as to provide more powerful services to the users.

In ArnetMiner, we define the researcher profile ontology and perform researcher profiling automatically using a unified approach. We integrate publications from the existing bibliography datasets. In the integration, we propose a constraints-based probabilistic model to deal with the problem of name disambiguation. We provide three types of search services. Moreover, we provide several mining services, such as expert finding, people association finding, and hot-topic finding.

The system advances four points: 1) proposal of a unified approach to researcher profiling, 2) proposal of a constraint-based probabilistic model to name disambiguation, 3) proposal of a score-and-propagate approach to expert finding, and 4) proposal of an efficient approach to association search.

2. System Overview

Figure 1 shows the architecture of the system. The system mainly consists of five main components:

1. *Extraction*: it automatically extracts the researcher profile from the Web by the following steps: 1) first collect and identify relevant pages (e.g. one's homepages or introducing pages) from the Web, 2) use a unified approach to extract the profiling information from the identified pages, and 3) collect publications from existing digital libraries.
2. *Integration*: it integrates the extracted researchers' profiles and the crawled publications. It employs the researcher name as the identifier. A constraint-based probabilistic model has been proposed to deal with the name ambiguity problem in the integration. The integrated data is stored into a researcher network knowledge base (RNKB).

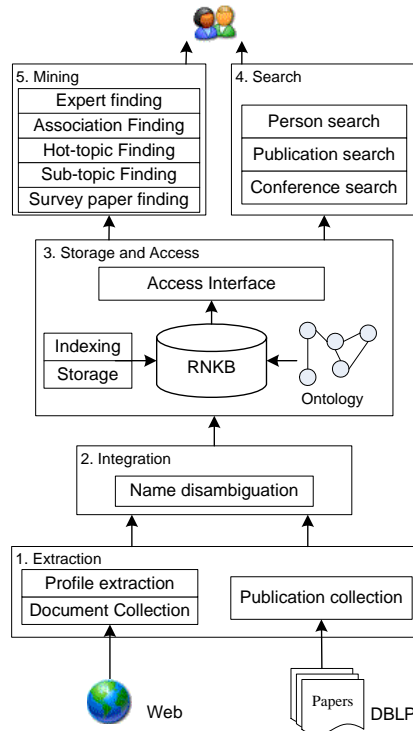


Figure 1. Architecture of ArnetMiner

3. *Storage and Access*: it provides storage and index for the extracted/integrated data in the RNKB. Specifically, for storage it employs Jena [2]; for index, it employs the inverted-file indexing method [9].
4. *Search*: it provides three types of search services: person search, publication search, and conference search. Given the name of a person, person search returns his/her profile information, authored publications, and relationships with the other researchers. Given a keyword, publication search returns the relevant publications. And conference search intends to find related conferences for a given keyword.
5. *Mining*: it provides five mining services: expert finding, people association finding, hot-topic finding, sub-topic finding, and survey paper finding. Given a topic, expert finding returns a list of persons who are 'experts' on the topic. Given a keyword, hot-topic and sub-topic finding returns the hottest research topics that researchers interested in and sub topics in that field. And given any two persons, people association finding returns possible associations between them. Survey paper finding is aimed at finding survey papers for a given topic, which is helpful for the researcher to gain a quick insight into a research topic.

For several features in the system, e.g., researcher profile extraction, name disambiguation, expert finding, and association search, we propose new approaches trying to overcome the drawbacks that exist in the conventional methods. For some

other features, e.g., storage, knowledge access, and searching, we utilize the state-of-the-art methods. This is because, these issues have been intensively investigated previously and the conventional methods can result in good performances. We also provides easy access interface (web services) for developing advanced applications.

Please note that this is a product of an ongoing project. Visitors should expect the system to change. We are extracting more researcher profiles and publications and are also developing more practical search services based on feedbacks from users.

3. Extraction of the Researcher Community

We define the researcher profile ontology (Figure 2), by extending FOAF [1]. In the ontology, two concepts, 24 properties and two object relations are defined.

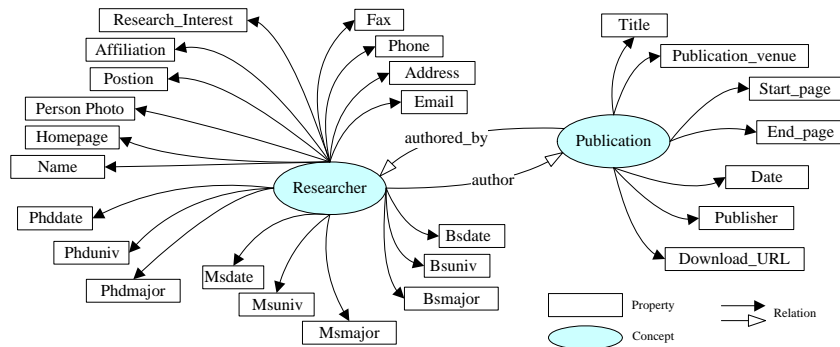


Figure 2. The researcher profile ontology

We randomly selected 1K researchers and studied how to extract profiles of the researchers from the Web. We found that it is non-trivial to perform the extraction. Specifically, we observed that 85.62% of the researchers are faculties of universities and 14.38% are researchers from company. For researchers from the same company, they might have similar template-based homepages. However, different companies have different templates. For researchers from universities, the layout and the content of the homepages vary largely depending on the authors. We have also found that 71.88% of the 1K Web pages are researchers' homepages and the rest are introducing pages. Characteristics of the two types of pages significantly differ from each other.

Statistical study also unveils that (strong) dependencies exist between profile properties. For example, there are 3,842 cases (12.98%) in our data that extraction of a property needs use the extraction results of the other properties. An ideal method should consider annotating all the properties together.

We propose a unified approach to researcher profiling [8]. The approach consists of three steps: relevant page identification, preprocessing, and tagging. In relevant page identification, given a researcher name, we first get a list of web pages by a search engine (we used Google API) and then identify the homepage/introducing page using a classifier. The performance of the classifier is 92.39% in terms of F1-measure. In preprocessing, (A) we separate the text into tokens and (B) we assign possible tags to each token. The tokens form the basic units and the pages form the sequences of units in the tagging problem. In tagging, given a sequence of units, we determine the

most likely corresponding sequence of tags by using a trained tagging model. (The type of the tags corresponds to the property defined in Figure 2.) In this paper, as the tagging model, we make use of Conditional Random Fields (CRFs) [4].

We conducted experiments to evaluate the performance of the unified approach. On the randomly chosen 1K researchers' pages, our approach can reach 83.37% (in terms of F1-measure) on average. We compared our method with several state-of-the-art methods, i.e., rule learning based method (Amilcare) and classification based method (SVM-based method). Our approach outperforms the two baseline methods.

4. Integration of Heterogeneous Data

We integrate the publication data from existing online data source. We chose DBLP bibliography (dblp.uni-trier.de/), which is one of the best formatted and organized bibliography datasets. DBLP covers approximately 800,000 papers from major Computer Science publication venues. In DBLP, authors are identified by their names. For integrating the researcher profiles and the publications data, we use researcher names and the author names as the identifier. The method inevitably has the ambiguity problem (different researchers have the same name).

The task of name disambiguation can be defined as follow: Given a person name a , we denote all publications containing the author named a as $P=\{p_1, p_2, \dots, p_n\}$. For each publication p_i , it has attributes: *title*, *conference*, *year*, *abstract*, *authors*, and *references*. Suppose there existing k actual researchers $\{y_1, y_2, \dots, y_k\}$ having the name a , our task is to assign these n publications to their real researcher y_i .

Our method is based on a unified probabilistic model using Hidden Markov Random Fields (HMRF) [8]. This model incorporates constraints and a parameterized-distance measure. The disambiguation problem is cast as assigning a tag to each paper with each tag representing an actual researcher y_i . Specifically, we define the a-posteriori probability as the objective function. We aims at finding the maximum of the objective function. We incorporate different types of constraints into the objective function, where constraints are considered as a form of supervision or background knowledge. If one paper's assignment violates a constraint, it will be penalized in some sense, which in turn affects the disambiguation result.

For evaluating the proposed disambiguation method, we created two test sets from the data collected in ArnetMiner. We applied our method to the two datasets and obtained 75% in terms of F1-measure. We compared our method with a baseline method using unsupervised clustering algorithm. The baseline is similar to that proposed by [6] except that [6] also use a search engine to help the disambiguation. Our method outperforms the baseline method by 8.0% in terms of F1-measure.

5. Storage and Access

ArnetMiner represents the data based on RDF/OWL and stores the extracted data in MySQL database using Jena, version 1.5 [2]. To query the data, we use SPARQL. We extracted about half million researcher profiles, integrated more than 0.8 million

publications, and extracted about 2.4 million co-author relationships between researchers with 5.38 relationships for each on average. We stored the data as RDF triples. In total, there are more than 10M N3 triples stored in the database.

For searching for instances with one property containing some keyword such as “Professor”, the naive SPARQL based method would not be efficient (sometimes even need use dozen of minutes). For efficiently performing this kind of search, we create an inverted-file index. Using the inverted-file index, we can efficiently search for the URI of the instances/properties that contain the keyword. Then we employ SPARQL to query the specified URI. In this way, the index-based method uses only 0.14 second to conduct an average search.

6. Search

In ArnetMiner, we provide three types of searches: person search, publication search, and conference search.

1. *Person search.* The user inputs a person name, and the system returns the profile of the person. We perform person search in the constructed researcher network. If a person can be found, the profile of the person stored in the local knowledge base will be displayed. The system also supports searching with constraints, for example, the user can input a query like “Jie Tang, aff:Tsinghua” to searches for the person “Jie Tang” and with its “affiliation” containing “Tsinghua”.
2. *Publication search.* The user inputs keywords, and the system returns publications with the most relevant publications on the top. We employ the conventional information retrieval model to do the publication search. Moreover, the system tries to find the download link of each publication from the web.
3. *Conference search.* The user inputs keywords (e.g. “ISWC 2006”), and the system returns the detailed information of the conference.

7. Mining

Currently, ArnetMiner provides five mining services: expert finding, people association finding, hot-topic finding, sub-topic finding, and survey paper finding.

7.1 Expert Finding

The goal of expert finding is to identify persons with some given expertise from the community: “Who are the experts on topic X in the researcher community?”.

We propose a new approach for finding experts in a web community in which we take into consideration of both person profile and relationships between persons. The approach consists of two stages, Candidate Scoring and Expert Propagation.

In **Candidate Scoring**, we use the person profile information to calculate an initial expert score for each person. The basic idea here is that if a person has (co)authored

many documents on a topic or if the person's name co-occurs many times with the topic, then it is likely that he/she is a candidate expert on the topic.

In **Expert Propagation**, we make use of relationships between persons to improve the accuracy of expert finding. The basic idea here is that if a person knows many experts on a topic or if the person's name co-occurs many times with an expert, then it is more likely that he/she is an expert on the topic.

Our intuition stems from our observations on how humans find an expert in the real world, namely by a) reading person profile information, and b) asking known experts to make a recommendation. Our approach is an implementation of the two observations by combining the person profile and the relationships in the Web community.

We conducted experiments to evaluate the method for expert finding. We assume that a real 'expert' is often active in the committees of the top conferences and organizations in his/her related research topics. We collected topics and answers (<http://keg.cs.tsinghua.edu.cn/project/PSN/dataset.html>). Experimental results show that our method outperforms the baseline method using only researcher profiles and the method using PageRank. See [10] for details.

7.2 People Association Finding

Given a web community, the people association is defined as a sequence of relationships $\{e'_{i1}, e'_{i2}, \dots, e'_{ij}\}$ satisfying $e'_{m(m+1)} \in E$ for $m=1, 2, \dots, l-1$, where v_i and v_j represents the source person and the target person, respectively.

Given a large-scale web community, to find all possible associations between two persons is obviously an NP-hard problem. In ArnetMiner, we concentrate ourselves on finding the most 'goodness' associations. We call the association with the smallest score (the small the best) as the *shortest association* and our goal is to find the *near-shortest associations*, whose scores are within a factor of $(1+\beta)$ of the score of the shortest association for some user-defined $\beta>0$. Our method consists of two stages.

1. Shortest association finding. It aims at finding shortest associations from all persons $v \in V \setminus v_j$ in the community to the target person v_j (including the shortest association from v_i to v_j with score L_{min}). We employed a heap-based Dijkstra algorithm to find the shortest associations between two persons.

2. Near-shortest associations finding. Based on the found shortest association score $L_{min}>0$ and a pre-defined parameter β , the algorithm requires enumeration of all associations that are less than $(1+\beta)L_{min}$ by a depth-first search. We constrain the length of an association to be less than a pre-defined threshold.

To evaluate the effectiveness of our proposed approach, we created 9 test sets. Experimental results show that our approach achieves high performance in all of the association search tasks. In terms of the average time, our approach can find associations in less than 3 seconds in most of the search tasks.

7.3 Hot-topic and Sub-topic Finding

Finding the hottest research topics and the sub topics in a research field is a very important issue. For sub-topic finding, a clustering algorithm is utilized to group the

papers that contain the keyword inputted by the user. A threshold is used to determine the number of clusters. Then each cluster is viewed as a sub-topic.

For hot-topic fining, we employ a language model based methods. It uses two steps: n-best Part-Of-Speech (POS) tagging and term (base-noun phrase) identification given the n-best POS-sequences. In the first step, it finds the n-best POS sequences for a sentence in the paper or paper title by estimating a language model from the training data. In the second step, it again uses a trained language model to estimate the best term sequence. For each term, a probability is assigned, representing its popularity. We view the terms with the highest probabilities as the hot topics.

7.4 Survey Paper Finding

A survey paper objectively surveys a body of previously published research on a topic, integrating information from several published papers. Researchers often start investigating a new research issue by first studying the survey papers of that field. We employed a classification based method to find the survey papers. Specifically, given a keyword, we use the state-of-the-art retrieval method to find a set of relevant papers and view the papers as candidates. Then we utilize a classification model to identify whether a paper is a survey paper or not. As the classification model, we employ Support Vector Machines (SVM). Features were defined in the classification model.

8. Experiences

Here, we share some thoughts about the strengths and the weaknesses of system.

Strengths

From experimental results, we see that ArnetMiner can achieve high performance in most of the key issues addressed, including profiling, integration, expert finding, and association finding. Some concluding remarks are as follows:

1) Automatic extraction of the researcher profile from the Web is feasible and the profile properties are usually inter-independent. By making use of the dependencies between the properties, the accuracy of the profile extraction can be improved.

2) Integration of data from different sources is necessary for web community. Name disambiguation is the key issue in the integration. Our approach based on HMRF model can obtain better results than the baseline method.

3) Efficient storing and access is very important for a Semantic Web application. Using the index-based method, the system can provide high efficiency in search.

4) Expert finding is an important issue in the academic community. The score-and-propagate approach can effectively combine the researcher profile and the relationships between researchers, and thus obtain high performance.

5) People association search is another important issue for searching web community. The proposed approach can efficiently find associations between people.

Weaknesses/Future works

1) Extraction of more types of relationship. In ArnetMiner, we use only the co-authorship as the relationship. In the future, we will extract other relationships, e.g., the relationship of co-organization and co-project etc.

2) FOAF file integration. In the current system, we use the Google search API to locate the pages and extract the profile from the identified page for a researcher. FOAF files are also important sources to get person description. We can further integrate FOAF files on the Web to get more information about the person.

As future work, we also plan to investigate more mining issues to empower the system, for example expertise publication finding and rising 'star' finding on a topic.

We received feedbacks from about one hundred users. Most of the feedbacks are positive. For example, some suggest that the expert finding approach is useful and it can be enhanced by adding several new features (e.g. reviewers finding for a paper). Some other feedbacks also ask for improvements of the system. For example, 5% of the feedbacks complain mistakes made in the profile extraction and 6.8% point out the integration mistakes (assigning publications to a wrong researcher). In addition, 5.5% of the feedbacks mention that the found research interests are not accurate and the method should be improved, which is also our current research issue.

9. Conclusion

In this paper, we have presented an expertise oriented search system, called ArnetMiner, for web community. We introduced the architecture and the main features of the system. We have described in detail the several issues that we are focusing on and proposed our approaches to them. We have carried out experiments for evaluating each of the proposed approaches. We also simply analyzed the strengths and weakness of the system.

References

- [1] D. Brickley and L. Miller. FOAF vocabulary specification, namespace document, September 2, 2004. <http://xmlns.com/foaf/0.1/>.
- [2] J.J. Carroll, J. Dickinson, C. Dollin, R. Reynolds, A. Seaborne, and K. Wilkinson. Jena: implementing the Semantic Web recommendations. In Proc. of WWW'2004, pp.74-83.
- [3] J. Golbeck. Web-based social networks: a survey and future directions. Technique Report.
- [4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In Proc. of ICML'2001, pp.282-289.
- [5] P. Mika. Flink: Semantic Web technology for the extraction and analysis of social networks. Web Semantics: Science, Services and Agents on the World Wide Web, 2005, v(3):211-223.
- [6] Y.F. Tan, M. Kan, and D. Lee. Search engine driven author disambiguation. In Proc. of JCDL'2006, Chapel Hill, NC, USA, June 2006, pp. 314-315.
- [7] J. Tang, M. Hong, J. Zhang, B. Liang, L. Yao, and J. Li. ArnetMiner: toward building and mining social networks. (Demo) In Proc. of SIGKDD'2007.
- [8] J. Tang, D. Zhang, and L. Yao. Social network extraction of academic researchers. In Proc. of ICDM'2007, to appear.
- [9] C.J. van Rijsbergen. Information retrieval. Butterworths, London, 1979.
- [10] J. Zhang, J. Tang, and J. Li. Expert finding in a social networks. In Proc. of Database Systems for Advanced Applications (DASFAA'2007).