DSN: A Knowledge-based Scholar Networking Practice Towards Research Community

Juan Zhao, Kejun Dong, Jianjun Yu, Wei Hu
Computer Network Information Center
Chinese Academy of Sciences
Beijing, China
{zhaojuan, kevin, yujj, huwei}@cnic.cn

***Abstract*—In this paper, we carry out a knowledge-based scholar network practice towards Research community, named Research Social Networking, shortly DSN, by setting up a large knowledge base of scientists. We discuss key technologies in the paper, including scholar disambiguation and relationship extraction with the better performance evaluation than traditional methods. The DSN system has been implemented and integrated with Duckling cloud service, known as ResearchOnline, with more than 60 thousand scientists and 100 thousand papers.**

Keywords-collaboration environment; social network; search engine; knowledge base; data-intensive

I. INTRODUCTION

Social networking service (SNS) has changed the way we interact and carry on with our daily lives. We are getting used to conducting communications, sharing information and expanding personal professional network through social networking service. Given the advantages of SNS in building relationships and information sharing, scientists and scholars can also use social networking to improve scholarship. As the development e-Science [1], researchers collaborate more often to each other globally via virtual communities. They need an SNS platform for knowing about what others in the field are thinking, finding more collaboration opportunities, and expanding the influence of their research achievements.

There have been several social networking sites targeting the group of scientists and scholars. ResearchGate [2] and Golaxy SocialScholar [3], like the Facebook in the scholarly community, offer users a platform to publish their papers and follow other scientists. Through these sites, scientists can get to know and connect with other scientists more easily. However, as scientists know about each other according to their academic background, the integrity and accuracy of users are more crucial. Besides, in their daily work, they have formed a fixed professional network. Recommending new scientists based on the existed network would be more trustful. So far as we know, many scholar social networking service ignore these aspects, and cannot reach good use effect.

In the progress of e-Science, much work has been done to facilitate scientists' information search and sharing, like scholar search engines such as Arnetminer [4] and Microsoft academic search [5]. Most scholarship search engines provide an advanced search service but are short of providing social networking functions, and thus, fail to encourage the communication. Although they lack the social gene, the social networking service can take advantage of the academic search, particularly the scholar-oriented search.

This paper presents a knowledge-based scholar networking service towards research community, called Duckling Social Networking, shortly DSN. The main unique characteristics of the DSN that distinguish it from other scholar social networking systems are as follows: 1) providing a knowledge base which extracting basic profiles, publications, news, projects and existing social networks of scientists from the web. 2) Ensuring the precision and integrity of the information with scholar disambiguation and specific extraction methods called RKLD, which is shown in this paper.

An implementation of DSN is integrated in a Research Internet community – ResearchOnline [6]. Users in the community can join DSN to communicate with each other. DSN also provides an API-based RESTful search service for other applications in ResearchOnline. We provide a mechanism for users in the community to be identified as and connect with the scientists in the knowledge base, so they can begin the social networking journey by searching scientists' works, following interested scientists and improving the social communication. With the support of knowledge base, the authority and integrity are assured and much academic knowledge is accumulated, including scientists' interactions and activities.

This paper reports the development of Duckling Social Networking. Some key technologies and DSN implementation also are discussed in the paper, including scholar disambiguation and the relationship extraction. In section II we discuss the related work on social networks in scholar community. In section III, we introduce the Duckling platform [7], as well as ResearchOnline. In section IV and V, we elaborate the architecture and workflow of Duckling Social Network, and focus on the key technologies. The implementation is given in section VI. Finally we discuss the conclusion and future plans in section VII and VIII.

II. RELATED WORK

For academic search, several research issues have been intensively investigated, for example expert finding and association search.

Expert finding is one of the most important issues for mining social networks. For example, both Nie et al. [8] and Balog et al. [9] propose extended language models to address

978-0-7695-5083-1/13 $25.00 © 2013 IEEE
DOI 10.1109/eScience.2013.20

the expert finding problem. From 2005, Text Retrieval Conference (TREC) has provided a platform with the Enterprise Search Track for researchers to empirically assess their methods for expert finding [10].

Association search aims at finding connections between people. Several researchers worked on extracting a social network from a community. Kautz et al., in 1997 [11] imagined ReferralWeb, a new system to extract relationship from the Web and email archive. The system focuses on co-occurrence of names on Web pages using a search engine. Matsuo et al. [12] created a system, called Polyphonet which recognizes four types of possible relationships between two actors. Adamic and Adar [13] have investigated the problem of association search in email networks.

In addition, a few systems have been developed for academic search such as, Google Scholar, Arnetminer and Microsoft academic search, some of which have tried to use the expert finding and association search techniques. Tangjie et al. [4] have developed Arnetminer, which is an author-based search engine extracting the profile of the author from the web and supporting the association search based on the co-author network. Microsoft academic search also provides advanced visualization applications such as co-author graph, paper-citation network and co-author path, which have been developed based on analysis of the co-author network. However, these systems are more like search engines which lack of the social networking service and the relation extraction mostly focused on the co-author relation.

As social networking service has swept the globe，there have been some social networking sites for scientists and researchers. For example, ResearchGate and Galaxy SocialScholar enable users to create professional profiles, discuss their work in topic specific Q&A forums, share papers, search for jobs and discover conferences in their field. But these systems focus more on social communication between scientists than providing less help on knowledge management. And they also lack the authorities of users' profiles, which is not helpful for encouraging collaboration. We solve this problem by setting up a large knowledge base. To the best of our knowledge, the issues we focus on in this work (e.g., profile extraction, scholar disambiguation, and relation extraction) have not been sufficiently investigated. Our system addresses all these problems holistically.

III. DUCKLING AND RESEARCH ONLINE

A. Collaboration Environment (Duckling)

The Collaboration Environment, supporting e-Science, is a comprehensive resource sharing and collaboration platform specific for research groups [14]. Via core software of the collaboration environment, including collaboration environment core toolkit and resource and service plug-in, all resources, such as hardware, software, data, information and human, can be organized and integrated together to form an efficient and easy-to-use system, supporting and advancing new research activity mode during the era of informationization. Duckling Collaboration Toolkit is a software suite supporting virtual organization, which can help to organize collaboration behavior and realize resource sharing and innovating.

As a development platform, Duckling Collaboration Toolkit supports standard portlet framework and has been used to integrate several discipline plug-ins to enable scientific research, including atmospheric data process, Matlab module analyze, and so on. Fig. 1 shows the components of Duckling software.

In general, Duckling 1.0 was released on Nov. 2008, as well as Duckling 1.1 in May. 2009 and Duckling 1.2 in Sep. 2009. Duckling 2.0, as an open source version, was released on Mar. 2010.

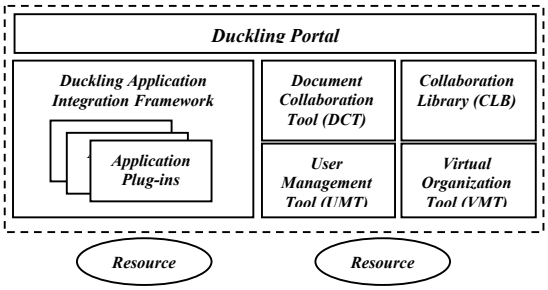

Fig. 1. Components of Duckling Platform

B. ResearchOnline

ResearchOnline is an integrated internet cloud service which is built on duckling platform and consists of sets of applications that are Conference Service Platform, Duckling Document Library and Duckling Social Network. The objective of ResearchOnline is providing a unified interface for researchers to carry on their research work. DSN is implemented as an application of ResearchOnline and serves users in the duckling community.

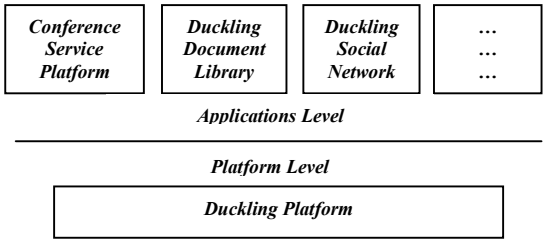

Fig. 2. ResearchOnline product line

IV. DUCKLING SOCIAL NETWORK

A. Architecture

Fig. 3 shows the architecture of DSNwith three layers:

1) Web Data Collection Layer

DSN uses a person-oriented crawler to fetch all kinds of data related to scientists from various deep web sources like literature database, search engines. In order to filter unrelated data, the fetched data are processed by a data preprocessor

through cleaning out all the unrelated pages that do not contain person names and the noise elements in page like images and links. And structured data can be extracted from web pages by wrappers to store in the local database [15]. Then we get the fundamental data such as structured databases, which are paper database, project database and scholar database which contains basic information such as names and affiliations about scientists, as well as unstructured data like web text.

2) Scholar Knowledge Formulation Layer

All the information about one scholar is scatted around the databases and the relationships between the scientists are hidden in the basic information. So we have Scholar Knowledge Formulation Layer, which is to integrate all the information about scientists from the fundamental databases into a scholar knowledge base. A scholar disambiguation framework is proposed to deal with problems of distinguishing and combining information about scientists, and relation extraction approaches are designed to extract the relationship between people.

3) API Layer

The API layer provides several services: scholar search, relation visualization and other social networking service. It also provides the WS-compatible RESTful data access interface.

B. Workflow

Fig. 4 illustrates the workflow of the scholar disambiguation and the relation extraction. Scholar disambiguation distinguishes the publications, projects and news to different persons and tags those with the related persons (Person models). The tagged publications, projects and news are then sent to the relation extraction module, which extracts three main types of scholar relationship, which are co-author, co-program and co-conference.

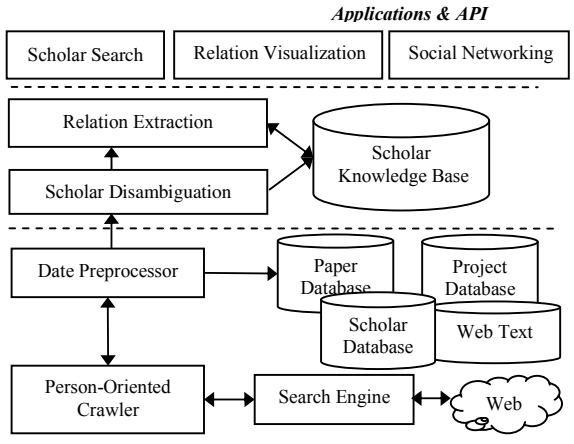

Fig. 3. Architecture of Duckling Social Network

V. KEY TECHNOLOGIES

This section illustrates two key technologies used in the DSN-scholar disambiguation and relation extraction, known as crucial components in the system architecture.

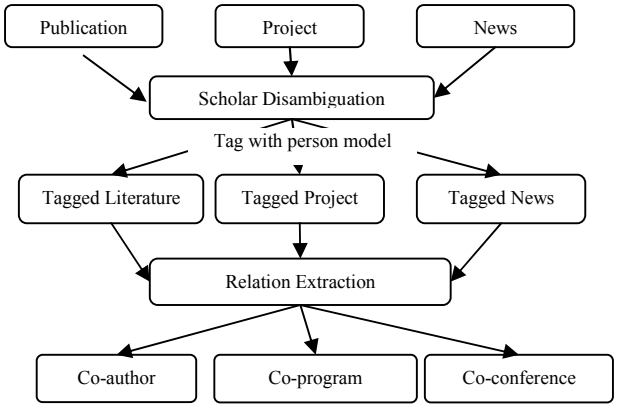

Fig. 4. User case and data workflow

A. Scholar Disambiguation

The task of scholar disambiguation is not only to distinguish different data to different persons with the same name, but also combine the different types of information to one single person, which is quite different from name disambiguation [15]. Our approach is based on the person model.

A person is a completed entity in the real world and cannot be identified with a name easily. To identify a person accurately, we use a person model mapping the person in the real world. Person model can be described by five dimensions/attributes: names such as the real name, nickname and alias names, which is denoted by N; basic attribute denoted by P, such as birth date, gender which cannot be easily changed; introductory attribute denoted by Q means the introductory information about a person, like affiliation, title, research interests and address; the keywords attribute denoted by R is the keywords in a context which person appears in. The identifying abilities of each dimension of the person model are a little different. The names N and basic attribute P are stable so if two persons with different N and P, they are not the same person.

The goal of the scholar disambiguation is to extract unique person models from various data. It contains three steps: person model attributes extraction, person model attributes identification, and person model clustering.

1) Person model attributes extraction.

For a web page, the first task is to extract and identify the attributes of people. Here we use the entity extraction techniques such as name and organization extractions. We use the ICTCLAS [17] combined with the name dictionaries to extract person names.

2) Person model attributes identification

When we extract several attributes on a page, we use names to identify the candidate person model. For example, if we extract three different names on a page, we have three candidate person models. The next is to decide which attribute belongs to which person model. For example, we may extract three candidate person models with three names, as well as other attributes like organization names, ages, and birthdates. We need to identify which age is to which person.

By observing a large amounts of web pages especially news, blogs, we found that person attributes usually appear around the names of describing a person. Based on the observation, we give the algorithm of people attributes identified in a text. The algorithm works as follows:

For each attribute p in attribute set P, calculate the distance between the positions of attribute p and the position of name n in the text, then the average distance $\overline{dis} = (N, P)$ can be calculated.

If the average distance is less than a threshold, the attribute p could be viewed as related with name n, and then p can be added to the person model n. The distance can be calculated in several different ways, such as the interval words, the product of the interval words and interval sentence when p and n are in different sentences.

We can use this algorithm for each web text to identify all the candidate attributes and fulfill a set of person models. However, as we extract person models per web page, there could be lots of same person names. We must distinguish person models of different persons and merge different models of the same person into one. So we come to the next step.

3) Person model clustering

We cluster person models based on computing the similarity of them. Given two person models X and Y, we can illustrate how the similarity is computed.

As we mentioned at the beginning of this session, the five dimensions of the person model have different abilities of identifying a person. Therefore we take different methods to compute different type attributes. For example, the basic attribute is stable and each attribute usually has one value. Assume that the two person models X and Y values in a dimension of the property P X_j = $\{x_j\}$, Y_j = $\{y_j\}$, so the similarity of the basic attribute can be computed by (1).

$$\mathrm{Sim}_P(X_j, Y_j) = \delta_P(x_j, y_j) = \begin{cases} 1, & x_j = y_j \\ 0, & x_j \neq y_j \end{cases} \quad (1)$$

For keyword attribute, which describes people with more information but is not as accurate as the basic attribute, we compute the similarity of keyword attributes with counting overlap of each word in (2).

$$\delta_Q(x_{ji}, y_{jk}) = overlap(x_{ji}, y_{jk}) = \frac{x_{ji} \cap y_{jk}}{x_{ji} \cup y_{jk}} \quad (2)$$

After computing the similarity of each type of attribute, we can combine them in (3). If the similarity Sim(X,Y) is larger than a threshold, X and Y are similar /close.

$$\mathrm{Sim}(X, Y) = \frac{\sum w \bullet \mathrm{Sim}(X_j, Y_j)}{N} \quad (3)$$

Here w is the weight of each type of attribute.

For clustering person models, we use the hierarchical clustering method.

Step1. The person model can be a cluster.

Step2. Choose two most similar clusters to merge into one single cluster.

Step3. If there is only one cluster left or the similarity between each cluster is less than a threshold, the process ends, or returns to step2.

4) Performance

We test our methods of distinguishing different person with same names. We extract different sizes of person models and count the overlap of person models. If three different persons with the same person are clustered into one person model, the overlap count is 2. The experimental result is listed in Fig. 5.

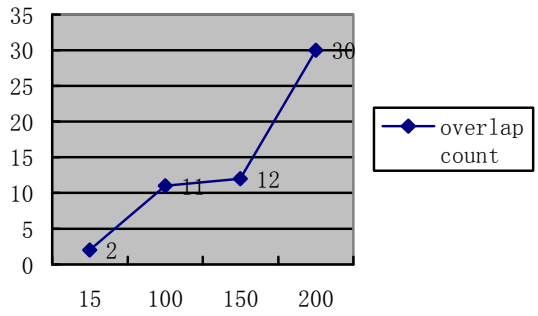

Fig. 5. Person model overlaps count

B. Relationship Extraction

We define more formally the problem of relationship extraction: extracting pairs of people who have the relationship in the relation set.

$$\forall pair = < p_1, p_2 > \in Pair, \exists PR_i(p_1, p_2) = true$$

Thus the relationship is extracted through two steps, extracting pairs of people who may be related, called people pair discovery, and identifying the relation type.

1) People pair discovery

In this step we need to find pairs of people who may be related. We use the co-occurrence approach. For each pair of persons X and Y, we compute the co-occurrence of them in their web documents. The co-occurrence function is defined as:

$$f(n_X, n_Y, n_{X \wedge Y}) = \begin{cases} \dfrac{n_{X \wedge Y}}{\min(n_X, n_Y)} & \text{if } n_X > k \text{ and } n_Y > k, \\ 0 \quad \text{otherwise} \end{cases} \quad (4)$$

$n_{X \wedge Y}$ can be estimated in several ways :

- co-occurrence of X and Y in a web document
- co-occurrence in the sliding window, such as a paragraph
- conjunction or punctuation concurrence, such as "X and Y attend the meeting"

This step generates the set of pairs $\{< P_x, P_y >, x \neq y\}$.

2) Relationship identification

In this step, we need to identify which type of relation the pair of people have and generate triples like $\{< P_x, P_y, PR_m >, x \neq y\}$.

The relationship identification can be viewed as a classification problem. Given a set of relations, determine which category the relation of two people belongs to. Therefore, we use the training text to learn classifier to classify the test text. We propose an advanced algorithm that is a keyword-based learning and semantic distance calculation method to construct the classifier (Relation Keyword Learning and semantic Distance computing, referred RKLD). RKLD can determine the type of relationship between entities. In RKLD, we define some concepts as follows:

- **Relation keyword: The** keywords that can describe the relation. The part of speech of a relation keyword is usually a verb.
- **Distance between relation keyword and person names:** Semantic distance between the position of the relation keywords in the context and the position of the two persons' names appears in. The smaller the distance, the stronger abilities of the relation keyword to describing the relation of the two people.

We illustrate the process and algorithm of the RKLD by giving this example: given two person models P_x , P_y, and the type of relation PR which contains some keywords K$\{k_i\}$. The task is to determine if P_x and P_y belong to PR. Suppose we have achieved related documents $T\{t_j\}$ about P_x and P_y.

RKLD is based on the algorithm of computing distance between relation keyword and person names. The algorithm works as follows: for each document t_j, get the context that both two persons' names appear in. For each keyword k_i in the type of the relation, compute the semantic distance between the relation keyword and the pair in the context $dis(k_i,t_j)$. As a name could appear several times in a document, the distance could be achieved by calculating the minimum, which is defined as:

$$dis(k_i, t_j) = \min(dis(k_i, P)) \quad (5)$$

Then, compute the mean distance:

$$mean(dis) = \frac{1}{m} \bullet \sum_{i=1}^{m} \left\| dis(k_i, t_j) \right\| \quad (6)$$

In calculation, the value of 0 indicates no relationship between keywords in the text fragment t_j. We assign it a large value which is much higher than the normal maximum range value.

For n documents, we have the final distance value:

$$Dis(P_x, P_y) = \frac{1}{n} \sum_{j=1}^{n} \text{mean}(dis) = \frac{1}{m \cdot n} \sum_{j=1}^{n} \sum_{i=1}^{m} dis(k_i, t_j) \quad (7)$$

The smaller $dis(P_x,P_y)$ is, the more P_x and P_y are likely to belong to this type of relation.

As RKLD is essentially a classifying process, it contains learning and testing steps. In learning step, we compute the $dis(P_x,P_y)$ for positive and negative documents to get the threshed φ, which is useful in determining the relation.

In testing step, we compute $dis(P_x,P_y)$ for each people pair and do the classification according to the rule as follows:

If $Dis(P_x, P_y) \leq \varphi$, then the pair belongs to relation PR; otherwise, the pair does not belong to relation PR.

3) Performance

To evaluate our method, we created a data set of 250 pairs of people as positives and 180 pairs as negatives. 4/5 of the data set are used for training and 1/5 of the data set are used for testing. For comparison purposes, we evaluate the results of RKLD and another classification method implemented by Support Vector Machine (SVM). We take the relation co-program as an example. We have two classes :{ co-program (CO), not co-program (NCO)}. Table 1 shows the results classified by RKLD and SVM respectively, and Fig. 6 shows the comparative statistics of the precision, recall, accuracy and error rate of RKLD and SVM. We see that our method outperforms SVM in precision, accuracy and error rate.

TABLE I. RKLD AND SVM RESULTS

	Truly CO（50）	**Truly NCO（36）**
Classify to CO (RKLD)	46	3
Classify to NCO (RKLD)	4	33
Classify to CO (SVM)	46	31
Classify to NCO (SVM)	4	5

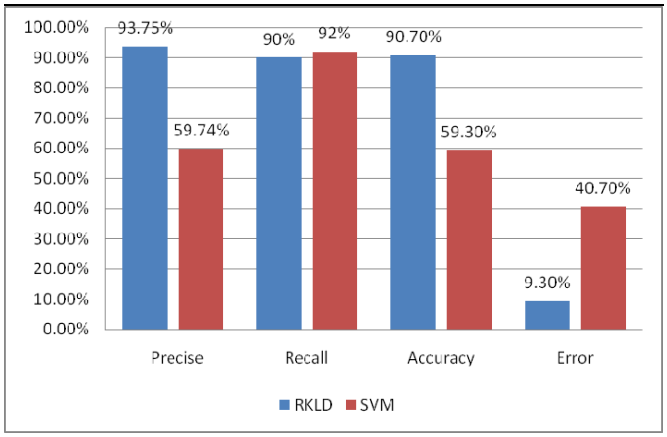

Fig. 6. Comparison on RKLD and SVM

VI. IMPLEMENTATION

We implement the key modules of DSN, essentially a person-oriented crawler, which fetched the basic profile of scientists from the national natural science foundation of China (NSF) [17]. The crawler is designed for collecting three types of information source including news, papers and projects, by fetching news from search engines like news.baidu.com, papers from ISI, ACM, WanFang [18] and projects from NSFC Library. An English-Chinese person name translation component is also implemented to help combine the English and Chinese publication of one scholar. So far, the crawler has fetched information about 60,000 Chinese scholars. All of the information fetched by the crawler is filtered and processed by the scholar disambiguation model and automatic extraction relations approaches, and stored in the final knowledge base.

Based on knowledge-based repository, a search engine is developed to facility scholar-oriented search, as a portal of DSN. A searched example is displayed in the DSN search interface to illustrate the UI of scholar-oriented search results (Fig. 7). The search engine classifies the searched results for a typed person's name into classes of the person page, paper, news and project. The system also supports scholars following other scholars, like social networking websites.

Fig. 8 is a web page that is tailored to an individual user, called scholar's homepage. The scholar's news, publications, projects, paper counting map, his/her fans and followers are shown along with the social network extracted from the Web. The left panel shows the relationship between the scholar and other related scholars, as well as a scholar relationship map which is developed by Flex, a RIA-based technology.

Once registered, a user's dashboard will be generated, which lists the latest feeds of concerns, follows and followers. The system also continues to recommend new scholars to users based on the extraction of the social network.

Fig. 7. DSN Search Interface

Fig. 8. DSN Scholar's Homepage

VII. CONCLUSION

In this paper, we introduce a knowledge-based social network practice towards research community, to facilitate the management of users' knowledge and the communication based on a social network extracted from the Web. We describe the architecture of DSN system and key technologies, i.e. scholar disambiguation and relation extraction. We propose a scholar disambiguation framework based on person model and a method named relation keyword learning and semantic distance computing to extract person relation. We conduct experiments for evaluating each of the proposed approaches. Experimental results indicate that the proposed methods can achieve a high performance. All of the approaches are applied into DSN. Finally, we demonstrate the implementation of the DSN.

VIII. FUTURE PLANS

There are still some improvements need to be done such as the interaction, the accuracy of the scholar disambiguation and relation extraction. DSN needs to integrate more relation types such as weak ties on Twitter and need to uncover more information hidden in the relationship such as what the program people collaborate, what conference they attend. Another interesting work is to find people's views towards conference, papers or even scholars.

ACKNOWLEDGMENT

This research was supported by NSFC Grant No.61202408 and CNIC Innovation Fund under Grant CNIC_CX_10004

REFERENCES

[1] Michael Jubb, Keith Adlam, e-infrastructure strategy, Report of the Working Group on Search and Navigation, pp.1,March 2006

[2] http://www.researchgate.net/

[3] http://soscholar.com/

[4] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, Zhong Su. 2008.ArnetMiner: extraction and mining of academic social networks. KDD'2008. pp.990~998.

[5] http://academic.research.microsoft.com/

[6] J. Yu, Y. Di, K. Dong, K. Nan, "Research Online: Cloud Service Platform for Internet Collaborative Environment," Journal of Huazhong University of Science and Technology (National Science Edition) Vol.39 Sup. I., pp. 33-37, June 2011.

[7] K. Nan, K. Dong, J. Xie, J. Yu, "Research collaboration platform for cloud service," Journal of Huazhong University of Science and Technology (National Science Edition) Vol.38 Sup. I., pp. 14-19, June 2010.

[8] Z. Nie, Y. Ma, S. Shi, J.-R. Wen, and W.-Y. Ma. Web object retrieval.In Proc. of WWW'07, pages 81–90, 2007

[9] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proc. of SIGIR'06*, pages 43–55, 2006.

[10] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec-2005 enterprise track. In *TREC'05*, pages 199–205, 2005.

[11] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of KDD'04*, pages 59–68,2004.

[12] D. M. Blei and J. D. McAuliffe. Supervised topic models. In Proc. Of NIPS'07, 2007.

[13] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27:187–203, 2005.

[14] Nan, K. Dong, J. Xie, D. Yang, J. Yu, VLAB: An e-Science Collaboration Framework for CAS Scientists, CODATA 2008, Springer Press, 2008.

[15] Zhao, Juan , Dong, Kejun; Yang, Le; Nan, Kai; Yan, Baoping ,E-Scholar: Improving academic search through combining metasearch with entity extraction, , Proceedings - 2009 IEEE Youth Conference on Information, Computing and Telecommunication, YC-ICT2009, p 247-250, 2009

[16] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proc. of WWW'05*, pages 463–470, 2005.

[17] Zhang, H.P., Yu, H.K., Xiong, D.Y., Liu, Q. HHMM-based Chinese Lexical Analyzer ICTCLAS. SIGHAN 2003, Association for Computational Linguistics (2003), 184-187.

[18] http://www.nsfc.gov.cn

[19] http://www.wanfang.com.cn