EFS: Expert Finding System based on Wikipedia Link Pattern Analysis

Kai-Hsiang Yang*, Chun-Yu Chen†, Hahn-Ming Lee†* and Jan-Ming Ho*
*Institute of Information Science, Academia Sinica,Taipei, Taiwan
Email: {khyang, hoho}@iis.sinica.edu.tw
†Department of Computer Science and Information Engineering,
National Taiwan University of Science and Technology, Taipei, Taiwan
Email: {m9515007, hmlee}@mail.ntust.edu.tw

***Abstract*—Building an expert finding system is very important for many applications especially in the academic environment. Previous work uses e-mails or web pages as corpus to analyze the expertise for each expert. In this paper, we present an Expert Finding System, abbreviated as EFS to build experts' profiles by using their journal publications. For a given proposal, the EFS first looks up the Wikipedia web site to get relative link information, and then list and rank all associated experts by using those information. In our experiments, we use a real-world dataset which comprises of 882 people and 13,654 papers, and are categorized into 9 expertise domains. Our experimental results show that the EFS works well on several expertise domains like "Artificial Intelligence" and "Image & Pattern Recognition" etc.**

***Keywords*—Expert Finding, Automatic Term Recognition, Wikipedia**

I. Introduction

Finding experts who have the appropriate skills and knowledge for a specify research field is an important task in academic activities. [1]. For example, editors of conferences or journals usually need to find appropriate experts to review submitted papers. The expert finding problem is traditionally solved by looking up expert-expertise databases. However, the databases are maintained manually and are really expensive to keep them up-to-date.

Several approaches are proposed in the literature. Balog et al. [2] proposed two models to find out experts from e-mail corpus, the first one is to locate the knowledge from the experts documents, and the second one is to locate the documents in the specify topic and then list all associated experts.

E-mail corpus naturally has a special feature, the communication link structure, and it could reveal social network relationships. Thus, several previous work [2, 3] uses some social network analysis methods to find out experts according to their social interactions.

Another particular consideration for the expert finding problem is that many research proposals are multi-disciplinary [4]. In our previous work [7], we adopted a language model to find out the experts. However, some limitations exist for the model, including the lack of supporting multidisciplinary search.

More precisely, an expert finding system usually focuses on the problem: "given a query topic and find out the experts who are familiar with it". In this paper we focus on a real world specific scenario: a science organization (NSC)[1], which have large amount of submitted proposals from researchers of all universities in Taiwan, wants to find experts to review those proposals. The committees of NSC need to assign all proposals to suitable experts in a short period of time. Before, experts are usually manually selected by the committees themselves. However, this causes some serious problems such as assigning a proposal to an expert who does not have enough expertise for that research topic. Besides, the manual process really takes a lot of time. As we mentioned above, many research proposals are multi-disciplinary, especially for the computer science domain. For example, some proposals may adopt data mining technologies to medical domains.

In this paper, we propose an expert finding system (EFS) that could provide: (1) building expert-expertise information automatically, (2) ranking experts according to their expertise relationship strength, and (3) providing the multi-disciplinary search. In the EFS, we use the experts' publications as the materials to build their expertise. An expert's publications can be treated as a strong evidence about his/her skills in academic field. In our scenario, using the publication corpus is more suitable than using other corpora (e.g. E-mail corpus) to find out experts. Moreover, we adopt the background knowledge to enrich the information of the experts' publications and the query proposals. Lots of work [8, 9] indicates that Wikipedia could improve the performance of the text categorization problem. Not only the content information, but also the link structures of Wikipedia we used. We believe that the link structure of Wikipedia describe the relationship of the expertise domains. It could help us to rank the experts precisely.

The proposed EFS comprises two parts, the expert profile building part and the user query part. In the first part, the EFS system uses candidate experts' publications to build an expertise profile for each expert. In order to represent the relationship strength of expertise, we adopt a predefined ontology, and the relationship in the ontology is used to search and rank the experts in the user query part. The system is based on linguistic and statically methods, where ontology is included to construct the expertise domain knowledge. To evaluate the effectiveness of the EFS, we use a real-world

[1] the Division of Computer Science of the National Science Council (NSC) in Taiwan

1-4244-2384-2/08/$20.00 © 2008 IEEE

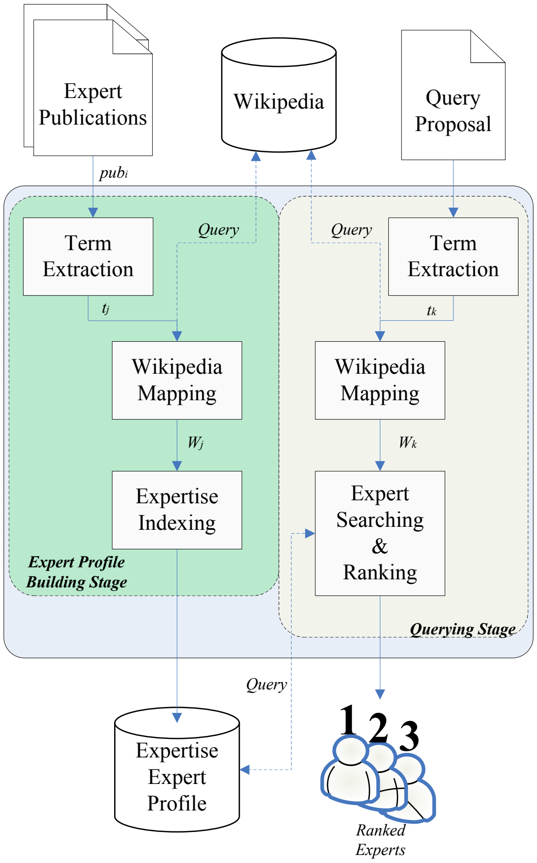

Fig. 1. System architecture of EFS

proposal allocation scenario to evaluate our system, instead of using predefined query terms in our previous work [7].

The remainder of this paper is organized as follows. In Section 2, we present the system architecture of EFS. Section 3 shows our performance evaluation for the proposed approach. The conclusions and future work is discussed in Section 4.

II. System Architecture

The main purpose of EFS is to find out experts who have enough expertise to review some given research proposals. The system architecture of EFS is shown in Figure 1. The input data for the EFS is a query proposal (in the up-right side), and the output is a list of ranked experts (in the down-right side). The EFS comprises two stages and four steps, including the expert profile building stage and the querying stage. Steps include the term extraction, Wikipedia mapping, expertise indexing and expert searching & ranking. Following sub-sections will describe all these stages and steps in detail.

A. Expert Profile Building Stage

In order to build the expertise-expert profile automatically, defining the material which can represent the experts' expertise domain first is important. Traditionally, the experts' profile describes his/her expertise domain in detail. However, not all contents in the experts' profile are suitable material to represent the experts' expertise domain. For example, some experts claim lots of expertise domains in their profile data. Therefore, the publications of experts' are selected. The publications are one of the strong evidences to represent ones expertise domain because these are reviewed by the other organizations and experts. Moreover, the publications are the well-known measurement in the academic communities.

In this stage, EFS builds the expertise-expert profile for each candidate experts. EFS uses the experts' publication title as the expertise evidence data. Not using the bag of word of the titles, term extraction process extract meaningful terms from the publication title.

To reinforce the expertise domain representation, the next step is Wikipedia mapping. Here the EFS maps the concept terms to the background knowledge, Wikipedia. Moreover, the Wikipedia elements of the terms are built by the link structure of Wikipedia. After this mapping process, the expertise domains of publications are represented by the Wikipedia elements.

In order to build the expertise-expert profile for EFS, Not only to address the expertise domains of candidate experts, but also to describe "Who are the experts of this expertise domain?" Therefore, the indexing of expertise domain and the experts is necessary.

B. Querying Stage

EFS performs the searching task by user's query proposals in this stage. The output of this stage is the ranked expert list. As like as the expert profile building stage, user's query proposals are proceed by the term extraction process to extract the terms. These terms then reinforce by the Wikipedia mapping process.

The last step is the expert searching & ranking step. EFS uses the Wikipedia elements of query proposals to match the expertise by querying the expertise-expert profile. It also ranks the experts according by the relatedness between their expertise and the query proposal.

C. Term Extraction

This section describes the term extraction process in detail. The expert profile building stage and the user query stage both contain this process. This process solves the question about how to group the words in the input string to meaningful terms. Traditionally, it is addressed by Automatic Terms Extraction (ATR) problem. EFS achieves it by the C-value method [5].

Just like the experts' profile, not all contents in the publication are suitable material to represent the experts' expertise domain. One publication has a title, an abstract, several content paragraphs and the citations. Some of the paragraphs and the citations contain farraginous information in the publication. For example, some of the paragraphs may describe the history of target topics and some of the citations may contain the preprocessing software they used. The title of publications has specific information against to the content paragraphs and the citations. Therefore, the title of publications is adopted.

There are three parts in the C-value method: part-of-speech tagging, linguistic filtering and statistical ranking. In the part-of-speech tagging part, EFS tags the part-of-speech of each words of the publication title. The Stanford Parser from The Stanford Natural Language Processing Group is adopted. This parser tags words by noun, verb, adjective, adverb and preposition.

After getting the tagging result, the linguistic filters fetch the candidate terms of each publications. Some candidate terms with special part-of-speech are not suitable to represent its publication titles, for example, the verb and the adverb. But the candidate terms with preposition like “of”, “for” have their function in some occasions. These are linguistic filters in the linguistic filtering part:

(1) $\text{Noun}^{+}\text{Noun}$

(2) $(\text{Adj}|\text{Noun})^{+}\text{Noun}$

(3) $((\text{Adj}|\text{Noun})^{+}|((\text{Adj}|\text{Noun})^{*}(\text{Prep})^{?})(\text{Adj}|\text{Noun})^{*})\text{Noun}$

Where the Adj. means adjective terms and the Prep. means preposition terms. We get the candidate terms of each publication through these three filters. Not all terms are useful and suitable, therefore, we call these terms are candidate terms. Some candidate terms are nested, for example, “fuzzy logic” is a nested term of “fuzzy logic control”. These nested terms are one of the features in the statistical ranking part.

EFS ranks the candidate terms from all result of linguistic filters in the statistical ranking part. C-value method ranks these candidate terms based on the term frequency and the number of times of the nested terms as equation (1).

C-value(term) =

$$\begin{cases} f(term) * log_2|term|, & \textit{term is not nested} \\ f(term) - \frac{NestedValue}{p(T_{term})}, & \textit{otherwise} \end{cases} \quad (1)$$

where

$$NestedValue = \sum_{nested \in T_{term}} f(nested) * log_2|term|.$$

In the above equation (1), the *f(term)* means the frequency of term, $|term|$ means the length of the term, T_{term} is the set of terms which contain term and $p(T_{term})$ is the number of the set of terms.

This method calculates each candidate terms by each linguistic method. In the linguistic filtering part, there are three types of candidate terms of each publication. We fetch the candidate terms which C-value is bigger than the average of the C-value in each linguistic type to be the representative terms of the publication. Table 1 shows the example of the representative terms of publication string “Neural-Network-Based Fuzzy Logic Control and Decision System”, the terms which C-value bigger than average will be selected. In this example, the average is 3.6794. The linguistic filter type is type 1

TABLE I
THE REPRESENTATIVE TERMS AND ITS C-VALUE

Representative terms	C-value	Selected
Fuzzy Logic Control	5.49306	v
Logic Control	5.54518	v
Decision System	0	

D. Wikipedia Mapping

EFS uses Wikipedia as the background knowledge source. Wikipedia has enormous of world knowledge and well defined knowledge structure. In order to use the Wikipedia as the background knowledge, EFS has to map the terms from the publications/proposals to the Wikipedia pages. In this process, EFS uses the search engine, Google, to do the term mapping. Google have the ability to search the query string in the specific web site.

EFS then follows the link structure of Wikipedia to build the Wikipedia elements of the concept terms. The following link structure of Wikipedia element includes:

(A) The Wikipedia page title

(B) The Wikipedia categories which contain (A)

(C) The Wikipedia categories which contain (B) as a child nodes

(D) The Wikipedia categories which contain (B) as a parent nodes

Figure 2 shows the Wikipedia element example. In this example, the Wikipedia element is extended by the page “Back Propagation”. In the Wikipedia page/category relationship structure, the term “Back Propagation” is included by the “Neural Network” category. Moreover, the “Neural Network” is included by the “Information, knowledge and uncertainly” and “Machine Learning” parent categories level. Even so, the “Machine Learning” category is included by the “Learning” category.

However, there are some limitations in the link structure of Wikipedia. For example, some of the Wikipedia categories are the internal tag of Wikipedia, like “Articles with unsourced statements since July 2007”. The internal categories are the noise of the Wikipedia element. EFS removes these categories manually.

One of the features of link structure in Wikipedia is the category scope. The scopes of the Wikipedia categories are unbalanced. The category “Neural network” contains 17 pages, and “Artificial Intelligence” category contains 35 pages. It indicates that the “Neural Network” expertise domain is more specific than the “Artificial Intelligence”. The unbalanced feature could be the ranking feature in EFS.

E. Expertise Indexing

Expertise indexing is the last step in the expert profile building stage. In the expert profile building part, the expertise domain profiles are built by each expert. It indicates that the expert-expertise profile describe “What the expertise does this expert have?” However, an expert finding system like EFS

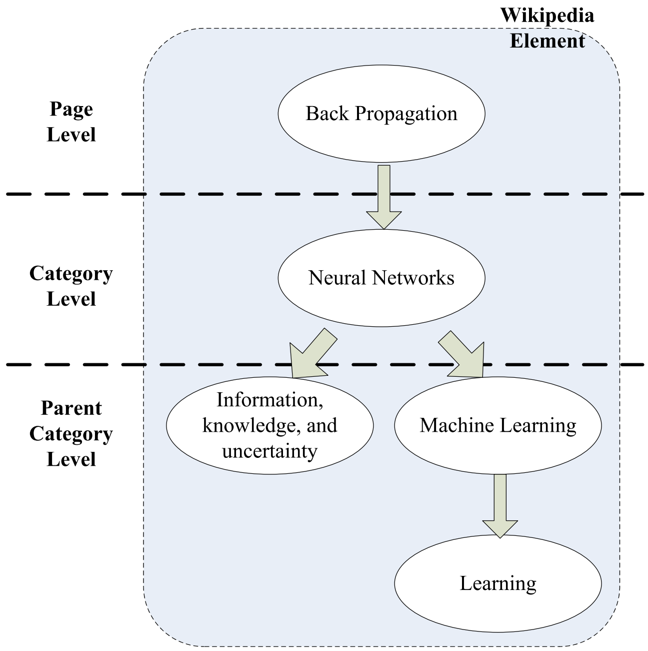

Fig. 2. The Wikipedia element example

should answers “Who are familiar about this kind of expertise?” or “Who are the expert of this expertise?” Therefore, EFS has to index the expertise domain to the experts’ names to achieve the requirement.

The index process depends on the structure of Wikipedia element. There are three levels in the Wikipedia element: the page, the category, and the parent/child category. Therefore, EFS indexes the concept terms by the page level, the category level, and the parent/child category level. In the ranking process, EFS could ranks the experts by their expertise domain depending on its level.

F. Expert Searching & Ranking

In the expert search process, EFS uses the Wikipedia elements of query proposal to search the indexed expert-expertise profile database. Depending on the three levels of Wikipedia elements, EFS searches the experts whose expertise match in the page level, the category level and the parent/child category level. The experts get different scores depend on the match levels.

EFS ranks the experts depend on the similarity difference between their expertise Wikipedia elements from expert-expertise profile and the Wikipedia elements of the query proposal. The following concepts are EFS followed:

(1) If two Wikipedia elements (expert’s expertise and the expertise of proposal) match in the page level, then the ranking score will bigger than match in the category level.

(2) If two Wikipedia elements match in the category level, then the ranking score will bigger than match in the parent/child category level.

(3) If two Wikipedia elements match in the category level, then the ranking score will cross the inverse of the times of the pages in match category.

(4) If two Wikipedia elements match in the parent/child category level, then the ranking score will cross the inverse of the times of the sub-category in match category.

III. Experiments

This section describes the experimental results. In order to evaluate EFS, we use the dataset which collected in our previous work [7]. This dataset includes 882 candidate experts who have submitted scientific proposals to the Division of Computer Science of the National Science Council (NSC) of Taiwan. It also collected the candidate experts’ publications; there are 13654 journal papers totally. In the NSC dataset, there are nine expertise domains and each of the candidates may have several expertise domains. The nine expertise are: (1) Image & Pattern Recognition (2) Natural Language & Speech Processing (3) Artificial Intelligence (4) Computer Graphics (5) Information System Management (6) Database (7) Bioinformatics (8) Web Technologies and (9) Quantum Computing. Moreover, the distributions of candidate experts with their expertise domains are list at Table 2.We use this NSC dataset in the expert profile building part.

For the user query part, we use 672 proposals which are submitted to the division of computer science of NSC in 2008. One of the nine expertise domains is assigned by the authors of the proposal in each proposal document. We use the expertise of these proposals to be the answer set. The distributions of proposals with its expertise domains list in Table 2. From the proposal distribution, it is easy to see that the largest amounts of proposals are the Image & Pattern Recognition domain and the Artificial Intelligence domain. So, we can say that these two expertise domains are more important than other expertise domains.

We simulated the scenario that mentioned in the section 1, the proposal allocation problem. The proposals are the input data, then, EFS search the suitable experts who are familiar the proposals expertise domain.

We measured EFS in terms of mean average precision (MAP), precision at 5, precision at 10 and R-precision. Table3 shows the results. From the experiment result, it is easy to see that our EFS has better performance at Artificial Intelligence, Image & Pattern Recognition and Natural Language & Speech Processing. In the Quantum Computing has worst performance. This may caused by rarely experts have Quantum Computing expertise in our dataset. Some of the expertise domains like Information System Management and Web Technologies has lower precision rate at 5 then the mean average precision. That means the ranking performance is not good enough in these expertise domains. The reason is the terms of these two expertise domains are too general and hard to distinguish with the other expertise domains.

IV. Conclusion

In this paper, we proposed an automatically system to find out experts who have enough expertise to review some given research proposals. Our system uses the experts’ publications to generate their expertise profile and then adopts ontology to

TABLE II
The expertise domain distribution of experts and proposals

Expertise domains	#Experts	#Proposals
(1) Image & Pattern Recognition	574	243
(2) Natural Language & Speech Processing	113	43
(3) Artificial Intelligence	722	115
(4) Computer Graphics	199	40
(5) Information System Management	720	30
(6) Database	318	68
(7) Bioinformatics	269	72
(8) Web Technologies	632	57
(9) Quantum Computing	8	3

TABLE III
Evaluation result of proposed EFS

Expertise domains	MAP	P5	P10	R-prec.
(1) Image & Pattern Recognition	0.39	0.47	0.47	0.50
(2) Natural Language & Speech Processing	0.36	0.44	0.33	0.29
(3) Artificial Intelligence	0.22	0.56	0.52	0.54
(4) Computer Graphics	0.3	0.28	0.26	0.17
(5) Information System Management	0.18	0.18	0.17	0.14
(6) Database	0.3	0.40	0.40	0.31
(7) Bioinformatics	0.18	0.33	0.33	0.43
(8) Web Technologies	0.28	0.23	0.26	0.55
(9) Quantum Computing	0.03	0.12	0.1	0.03

organize the expertise knowledge. In our experiments, we use a real world dataset from the NSC in Taiwan, and our experimental results show that our EFS works very well on some expertise domains like “Artificial Intelligence” and “Image & Pattern Recognition” etc. We use the experts’ publications as the evidence to their expertise. However in fact, the importance for each publication is not exactly the same. Some publications are very important in the specific domain and their authors should have more influence weights than others. The simplest way to solve this problem is to use the Impact Factor for ranking the publications. But it is hard to compare the Impact Factor between different expertise domains, and we can not use an average Impact Factor value to judge publications. Therefore, in the future we plan to add link structures between the publications into our system, because the link structure between the publications refers to the citation information, and this would be very useful to mine publications that are more influential than others by analyzing the citation relations.

Acknowledgment

This work was supported in part by the National Digital Archive Program (NDAP, Taiwan), the National Science Council of Taiwan under grants NSC 96-2628-E-011-084-MY3, NSC 96-2221-E-011-064-MY3, NSC 95-3114-P-001-002-Y02, NSC95-3114-P-001-001-Y02 and NSC 95-2221-E-001-021-MY3.

References

[1] Text REtrieval Conference (TREC). Enterprise Track 2005. [Online].Available: http://www.ins.cwi.nl/projects/trec-ent/wiki/

[2] Krisztian Balog and Maarten de Rijke, “Finding experts and their details in e-mail corpora,” *in Proceedings of the 15th International Conference on World Wide Web*, 2006, pp. 1035-1036.

[3] Christopher S. Campbell, Paul P. Maglio, Alex Cozzi, and Byron Dom, “Expertise identification using email communications,” *in Proceedings of the 12th ACM Conference on Information and Knowledge Management*, 2003, pp. 528-531.

[4] Seth Hettich and Michael J. Pazzani, “Mining for Proposal Reviewers: Lessons Learned at the National Science Foundation,” *in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 862-871.

[5] Katerina Frantzi, Sophia Ananiadou, Hideki Mima, “Automatic recognition of multi-word terms: the C-value/NC-value method,” *International Journal on Digital Libraries*, 2000, pp. 115-130.

[6] Arjen P. de Vries, Ian Soboroff, “Overview of the TREC-2006 Enterprise Track,” TREC 2006 Conference Notebook

[7] Chia-Ching Chou, Kai-Hsiang Yang, Hahn-Ming Lee, “”AEFS: Authoritative Expert Finding System Based on a Language Model and Social Network Analysis,” *TAAI* 2007

[8] Somnath Banerjee, “Boosting Inductive Transfer for Text Classification Using Wikipedia,” *in Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA)*, 2007 , pp. 148-153.

[9] Evgeniy Gabrilovich and Shaul Markovitch, “Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge,” *in Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006, pp. 1301-1306.

[10] Evgeniy Gabrilovich and Shaul Markovitch, “Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis,” *in Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2006, pp. 1301-1306.

[11] Dawit Yimam, “Expert Finding Systems for Organizations: Domain Analysis and the DEMOIR Approach,” *Journal of Organizational Computing and Electronic Commerce*, 2003

[12] Krisztian Balog, Leif Azzopardi, Maarten de Rijke, “Formal models for expert finding in enterprise corpora,” *in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 43-50.

[13] David Mimno and Andrew McCallum, “Expertise Modeling for Matching Papers with Reviewers,” *in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 500-509.