



HAL
open science

An Inner-Enterprise Wiki System (IWkS) Integrated with an Expert Finding Mechanism for Lesson-Learned Knowledge Accumulation in Product Design

Ywen Huang, Zhua Jiang, Xde Xiang, Cneng He, Jfeng Liu, Ying Huang

► To cite this version:

Ywen Huang, Zhua Jiang, Xde Xiang, Cneng He, Jfeng Liu, et al.. An Inner-Enterprise Wiki System (IWkS) Integrated with an Expert Finding Mechanism for Lesson-Learned Knowledge Accumulation in Product Design. 13th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Nov 2014, Ho Chi Minh City, Vietnam. pp.198-208, 10.1007/978-3-662-45237-0_20 . hal-01405582

HAL Id: hal-01405582

<https://inria.hal.science/hal-01405582v1>

Submitted on 30 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

An inner-enterprise wiki system (IWkS) integrated with an expert finding mechanism for lesson-learned knowledge accumulation in product design

Ywen Huang¹, Zhua Jiang¹, Xde Xiang², Cneng He², Jfeng Liu², Ying Huang¹

¹School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

²Institute of Product Design, Shanghai Waigaoqiao Ship-building Company, Shanghai, China

hywen2004@sjtu.edu.cn, zhjiang@sjtu.edu.cn

Abstract. In the process of product design, it is usually difficult for enterprises to timely and effectively accumulate lesson-learned knowledge, which should be revised several rounds by appropriate experts in a cooperative, multidisciplinary team to make sure of its accuracy and integrity. This study proposes a model of an inner-enterprise wiki system (IWkS) integrated with an expert finding framework to support the accumulation of lesson-learned knowledge in product design. By combining wiki's characteristics, the expert finding framework considers both users' expertise relevance, and social importance in IWkS. To validate our expert finding approach, some experiments are done in a famous ship-building company in China. Meanwhile, a working scenario of the IWkS integrated with an expert finding framework for knowledge accumulation is shown.

Keywords: Knowledge management, wiki, expert finding, product design.

1 Introduction

In recent years, more and more enterprises begin to realize that knowledge is extremely important [1], and management of knowledge resources is an essential way to enhance business efficiency and competitiveness [2]. Product design is a knowledge-intensive activity in manufacturing enterprises [3]. In the process of product design, engineers usually write lesson-learned documents for empirical knowledge sharing to others, and a lot of empirical knowledge is in the forms of lesson-learned documents (see Fig.1). As the task in product design is usually a

cooperative, multidisciplinary team work, the lesson-learned knowledge is collaborative and combined with the individual intelligence of team members. So when an engineer creates a rough lesson-learned item, the enterprise wants it to be revised timely and correctly in several rounds to reach a high-quality level by appropriate experts. The aforementioned condition has become more feasible by the advent of Web 2.0 systems, such as wikis, which provides a mechanism for collective knowledge creation and sharing [4]. The characteristics of wikis, like openness, simplicity and ease of maintenance, have made them be adopted by enterprises [5]. However, the quality of the collectively created knowledge in wikis is usually not very high. An expert finding mechanism is therefore needed to actively select the most appropriate expert peers in the organization to improve the content and ensure the quality. In this paper we propose an inner-enterprise wiki system (IWkS) integrated with an expert finding mechanism to support lesson-learned knowledge accumulation in product design.

The expert finding technique has attracted attentions of many research groups in various contexts, such as organizations and online communities [6, 7]. Existing models mainly utilize three sources of expertise indicators to make expert recommendations: self-statement information [8], authored documents [6, 9], and social interaction history [10, 11]. Furthermore, very few studies focus on virtual communities, especially wikis, where information quality is often not so good as that in organizational knowledge repositories. In this research, a hybrid expert finding approach is proposed and integrated into an inner-enterprise wiki, which evaluate users' expertise considers both contents of authored items (created or revised) and one's social importance within IWkS. Experiments have been conducted to assess and evaluate the effectiveness and usability of the proposed expert finding mechanism in the real world.

The rest of the paper is organized as follows. IWkS for knowledge accumulation is introduced in the next section. Section 3 presents a novel expert finding mechanism. Some experiments are done in Section 4 to validate our expert finding approach and a working scenario of the whole IWkS is shown in Section 5. Conclusions are then outlined in the last section.

2 IWkS for lesson-learned knowledge accumulation

2.1 Lesson-learned knowledge

In product design, lesson-learned knowledge represents the skill-related knowledge acquired from an engineer's past experience in performing design tasks. A lot of such knowledge is generated, which makes it possible for an engineer to reuse others' experience in accomplishing various tasks. In many enterprises without knowledge management tools, the lesson-learned knowledge is in the form of documents which are stored in the local network. Fig.1 shows content of a lesson-learned document on outfitting design of a ship-building company which is written by the design department, and each document has main parts: item title, problem description, method description, etc.


Document ID [Ⓢ]	WD2012060302 [Ⓢ]
Document Name [Ⓢ]	Structure modeling [Ⓢ]
Problem Description: [Ⓢ] In the structure modeling of ship H1115, the phenomena of interference among pipes and structures usually happen. [Ⓢ]	
Method Description: [Ⓢ]  When doing structure design, distances among structure models should be controlled. Theoretically, as long as the distance between two structures is more than 20mm, it's OK for structure modeling. However, the production capacity of our company couldn't meet that requirement, and the distance should be larger than 50mm for structure manufacturing. [Ⓢ] So when designing structures and checking interferences among structures, we cannot just only obey to the standard files, we also have to know about the production capacity. What more, engineers of different majors should strengthen communication. When modeling, if space allows, the model spacing should be as large as possible. [Ⓢ]	
Author ID: Engineer024 [Ⓢ]	Author Name: Cheng HE [Ⓢ]

Fig. 1. Content of a lesson-learned document

2.2 Mechanism of knowledge accumulation through IWkS

With a wiki system, the enterprise can have a better knowledge accumulation mechanism for engineers to create and accumulate lesson-learned knowledge. In Fig. 2, an overview of procedures of knowledge accumulation through IWkS is illustrated. After a new rough lesson-learned item is created and submitted into IWkS, or an existing item is revised, the item will be peer reviewed to assess its quality. Next, if quality of an item is inadequate, it is necessary to find the most appropriate engineer to improve the specific revise-to-be item, which is controlled by the expert finding mechanism. The details of the proposed expert finding approach are introduced later in Section 3.

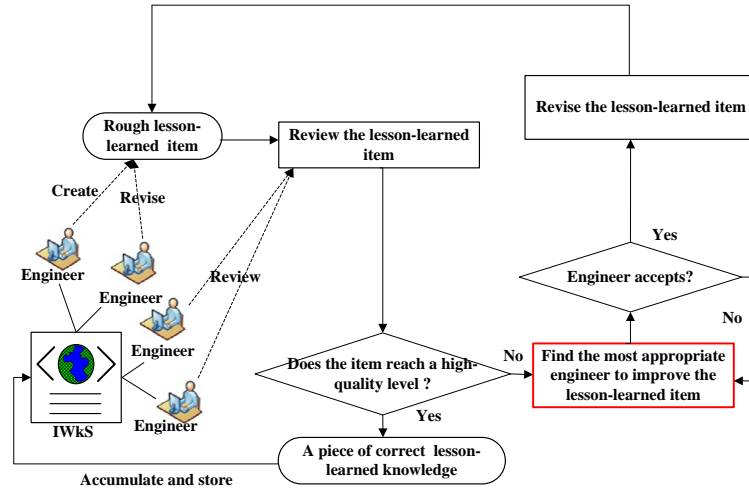


Fig. 2. An overview of knowledge accumulation through IWkS

As soon as a peer engineer, selected as an appropriate expert by the proposed algorithm, gives revisions to the lesson-learned item, it undergoes a re-assessment. The revision, review and expert finding processes can be repeated several times until the rough lesson-learned item has surpassed the quality threshold. Lesson-learned items, which reach a high-quality level, will be shared and reused among all the engineers in the future product design.

3 Expert finding approaches in IWkS

3.1 Framework for the expert finding mechanism

In IWkS, there are many lesson-learned items in the domain of product design. Engineers are encouraged to create or give different revisions and comments on their familiar items to improve the quality of lesson-learned knowledge in IWkS. Whether an engineer can be selected as the potential appropriate expert on a certain topic depends not only on whether the engineer has relevant knowledge, but also on the social importance or authority in IWkS community.

Therefore, the expert rank $ExpertRank(engr_i, RI_k)$ of an engineer $engr_i$ for a certain rough lesson-learned item RI_k is defined as a combination of expertise relevance and social importance, which is shown in Formula (1).

$$ExpertRank(engr_i, RI_k) = a \cdot ER(engr_i, RI_k) + (1 - a) \cdot SI(engr_i) \quad (1)$$

Where $ER(engr_i, RI_k)$ denotes expertise relevance score between an engineer's expertise and a rough lesson-learned item RI_k ; $SI(engr_i)$ denotes social importance score of the engineer $engr_i$, reflecting authority of the engineer within the IWkS community; and $a \in [0, 1]$ is a coefficient of the linear combination strategy that combines expertise relevance and social importance into a single expert rank score. As shown in Fig.3, the framework of an expert finding mechanism in IWkS is proposed that attempts to detect experts from IWkS community, and has four main phases.

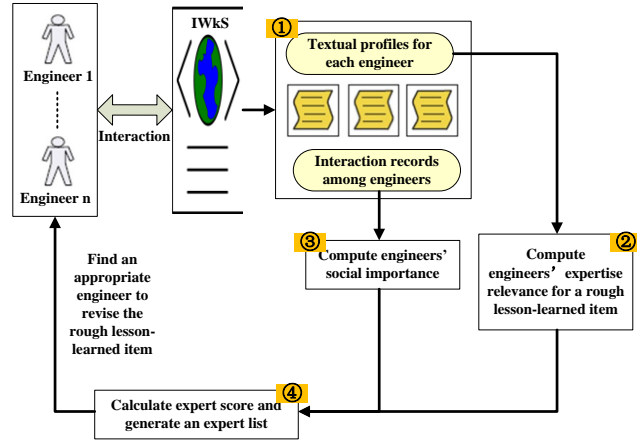


Fig. 3. The framework of expert finding mechanism

① Build textual profiles for each engineer and collect interaction records among engineers from IWkS. The textual profiles are used to extract expertise for each engineer. The interaction records among engineers are used to calculate engineers' social importance. ② Compute relevance between each engineer's expertise and a rough lesson-learned item. The details are introduced in Section 3.2. ③ Compute engineers' social importance. The details are elaborated in Section 3.3. ④ Compute expert rank score. In this phase, a hybrid method (see Formula 1) is used and a recommended expert list is generated.

3.2 Expertise relevance computation

Expertise extraction. Each engineer's textual profile is built by merging all the items in IWkS that the engineer has previously created or revised. In general, specific text preprocessing is required before automated topic extraction. Once these pre-processes

are finished, for a textual profile, some meaningless and superfluous words are filtered out, and some words are cut down to the root by a stemming algorithm[12]. Then each topic will receive a weight using a term weighting algorithm, which is based on the Term Frequency/Inverse Document Frequency (TF/IDF) method[13]. Once topics are ordered by weights, each engineer's expertise is denoted by a set of tuples (t, E_t) , where t is the topic and E_t is its weight.

Relevance between each engineer's expertise and a rough item. Once the topics of expertise are decided, the main issue is to calculate relevance score. Expertise relevance indicates whether an engineer in IWkS community has any level of expertise with regard to a rough lesson-learned item in product design. The rough item can also be represented as several topics and their weights (t, I_t) by using TF-IDF algorithm [13].

Then we can calculate an expertise relevance score between an engineer's expertise and a rough lesson-learned item by Formula (2), which is deployed to Formula (3), (4) and (5).

$$ER(engr_i, RI_k) = \frac{sim(EE_i, RI_k) \times \sum_{j \in C_{engr_i}} E_j \times T_j}{|C_{engr_i}|} \quad (2)$$

Where EE_i denotes expertise of engineer $engr_i$; RI_k denotes the k th rough lesson-learned item; C_{engr_i} denotes the textual profile of engineer $engr_i$; j denotes the j th item in C_{engr_i} ; $|C_{engr_i}|$ denotes the number of items in C_{engr_i} and $|C_{engr_i}| \geq 1$.

$$sim(EE_i, RI_k) = \frac{\sum_{t=1}^m E_t I_t}{\sqrt{\sum_{t=1}^m E_t^2 \sum_{t=1}^m I_t^2}} \quad (3)$$

Where E_t denotes the weight of topic t in an engineer's expertise EE_i ; m is the total number of topics in EE_i and I_t denotes the weight of topic t in the rough lesson-learned item RI_k .

$$E_j = 1 + \frac{n(\text{positive evaluation}) - n(\text{negative evaluation})}{n(\text{evaluation})} \quad (4)$$

E_j is the evaluation result of lesson-learned item I_j , which is used as a factor to discriminate the quality of revisions made by the contributors. If someone thinks the revised lesson-learned item is low-quality level, he or she can give it a

positive/neutral/negative evaluation, which is considered as plus, zero, and minus, respectively. n denotes the number of evaluations received and $n \geq 1$. To prevent the value of evaluation factor being lower than 0, we add one to the result.

$$T_j = e^{-\tau(t_{now} - t_j)} \quad (5)$$

The next factor considered is the time factor. It is reasonable to assume that the version of a revised item posted in the recent past is more important and up-to-date. The time weight of each version of item is calculated as Formula (5). Where t_{now} is the current date time; t_j is the date time of the item I_j being revised; τ is an adjustable parameter, which we set at $1/365$ to avoid dropping too fast [14].

3.3 Engineers' social importance computation

Social network construction. To determine an engineer's social importance in IWkS, firstly a social network should be constructed to represent interactions of engineers in the IWkS community. In IWkS, when an engineer initiates a lesson-learned thread by creating a new item on a certain topic, others may follow this thread and modify the item to get a new version. The engineer-item relationships are illustrated in Fig. 4(a). There are two sorts of vertices in the graph: engineer enr_i and item thread I_k . The directed edge from an engineer to an item means that the item is initialized by the engineer. The directed edge from an item to an engineer means the same item is revised by the engineer. Engineer-engineer relationships are extracted through connecting engineers who participate in the same thread with directed edges from item creators to revisers: see Fig. 4(b). Social interactions are depicted in the engineer-engineer graph, which is used to infer an engineer's social importance in IWkS.

In the knowledge accumulation context of IWkS, according to social network theory, an engineer A will receive a vote of support from another engineer B whose initial rough item is revised by A. If several engineers give revisions to modify B's thread, the vote of support from B is evenly distributed to those engineers. Each element in the engineer-engineer relationship matrix U , can be calculated as Formula (6).

$$u_{ij} = \begin{cases} 0 & \text{if } (i, j) \notin E \\ 1 / out(i) & \text{if } (i, j) \in E \end{cases} \quad (6)$$

Where $out(i)$ is engineer enr_i 's out-degree, i.e., the total number of engineers who have helped engineer enr_i to revise the item. E is a set that consists of directed links from item thread creators to revisers. Each element u_{ij} represents the vote of

support received for engineer $engr_j$ from engineer $engr_i$.

Take the engineer-engineer relationship graph in Fig. 4(b) as an example,

$$u_{ij} = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$



Fig. 4. An engineer-item graph and an engineer-engineer relation graph

Social importance computation. The social importance score represents a user's authority in an engineer-engineer relation graph. PageRank algorithm is used to calculate social importance score, which is shown in Formula (7).

$$SI(engr_i) = d \cdot \sum_{j=1}^N SI(engr_j) \cdot u_{ji} + (1-d) \frac{1}{N} \quad (7)$$

Where d is a damping factor, it allows the recursive cycles to jump to a random engineer in IWkS. It is commonly set to 0.85 [11]. An iterative algorithm, like the power iteration method, can be used in the recursive computation. The initial value of social importance score $SI(i)$ can be given as $1/N$ [11]. The iteration ends when the social importance scores converge.

4 Experiments

Since the proposed expert finding mechanism integrated in IWkS is important for lesson-learned knowledge accumulation, it should be evaluated. So the experiment in this section is conducted to investigate (1) the performance influenced by changing the parameter a in Formula 1; and (2) the effectiveness of the proposed method.

4.1 Experimental design

SWS is in a famous shipbuilding company in China. The engineers in the field of outfitting design are asked to use a traditional wiki system to share their lesson-learned knowledge. They are encouraged to give contributions and reviews for lesson-

learned items. Moreover, technical consulting committee has manually identified the experts in the wiki community. Each quarter several distinguished engineers are selected to receive the Special Contribution Award for their enthusiasms and capability to improve the quality of the lesson-learned items. The historical identified expert lists can be naturally employed as the gold standard for the empirical evaluations. All the versions of items and interaction records among engineers in the wiki system from 2012 to the end of year 2013 are crawled, with approximately twenty eight projects.

Several information retrieval metrics are employed to evaluate our experimental results including precision (P), recall (R), macro-average F-measure (Ma), micro-average F-measure (Mi), Mean Reciprocal Rank (MPR), Mean Average Precision (MAP) and P@5 [11].

4.2 The optimal parameter value

The first experiment is to determine the optimal parameter a in the linear combination strategy. We try different a to compute the final expert rank score in Formula (1). a is the parameter used to adjust the relative importance of expertise relevance score and social importance score. Fig. 5 shows the top-5 expert ranking results based on expert rank scores. For P@5 and MAP, the performance increases when a rises and peaks at $a = 0.7$, so the best performance is achieved by setting $a = 0.7$. The expertise relevance score is more important than the social importance score to derive the final expert rank score.

4.3 The effectiveness of the proposed method

The second experiment is to evaluate the effectiveness of our expert finding approach by comparing with a baseline method, where only the content (textual profile)-based expertise relevance is considered to find the experts. And the parameter a value is set to its optimal value 0.7. The experimental results are shown in Fig. 6. The proposed approach significantly outperformed the content-based baseline method ($p < 0.001$). The proposed approach's F-measures (MacroF1 and MicroF1) are around 36%, while those of the baseline method are only about 12%.

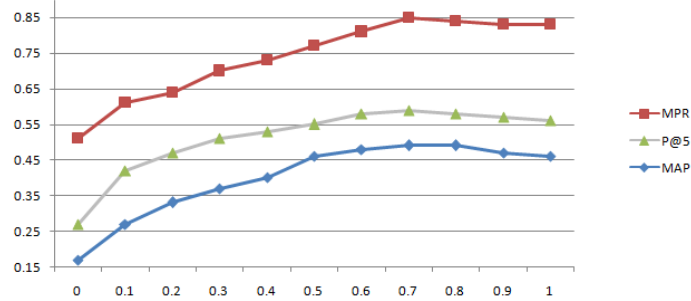


Fig. 5. Expert finding performance using different a

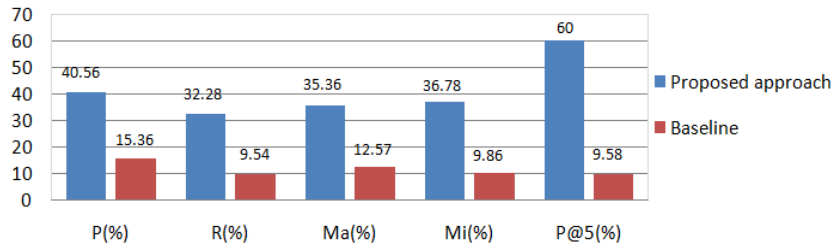


Fig. 6. Performance comparison between proposed approach and a baseline one

5 A working scenario of the whole IWkS for knowledge accumulation

According to those considerations, we have designed an operational wiki system IWkS integrated with an expert finding mechanism for lesson-learned knowledge accumulation in outfitting design. The system framework including the knowledge accumulation function is realized by MediaWiki technique, and the expert finding mechanism is implemented by the method proposed in Section 3. As shown in Fig.7, when a rough lesson-learned item submitted into IWkS, by using our expert finding mechanism, the appropriate expert list will generate according to the expert rank. Then IWkS will send a system request to the top one expert to ask him or her give a revision on the item. The expert finding process can be executed by the system automatically. In some specific circumstances, the manager can use the “expert finding” button to find experts for certain items manually.

Outfitting Design
IWkS

Project page Discussion Read Edit Add topic View list **Revise-to-be item** Go Search

Item title: Interference between outfitting pieces and pipe

Problem Description: In the building process of Ship H1115, phenomena of interferences between pipelines and outfitting pieces usually happen...

Method Description: When doing the production design, theoretically, the distance between pipelines and outfitting pieces is 20mm, however, as the capability of our factory is limited, the distances among three - dimensional models should at least be 50mm...

Expert finding

	ID	Name	Department	Rank
	74206	Wangying	Electrical outfitting	1
	73458	XiongYangbin	Electrical outfitting	2
	73982	HuangYamei	Mechanical outfitting	3
	74239	LuoRongshun	Electrical outfitting	4
	73928	HuangZhongbin	Electrical outfitting	5

The top 5 appropriate expert

Fig. 7. Expert finding results for a revise-to-be lesson-learned item in outfitting design

6 Conclusions

This study proposes an inner-enterprise wiki system (IWkS) integrated with an expert finding mechanism to support the accumulation of lesson-learned knowledge in product design. By combining wiki's characteristics, the expert finding mechanism is a hybrid approach considering both users' expertise relevance, and social importance in IWkS. Experiments show that the proposed approach achieves better performance than the baseline one. The promising results may confirm the feasibility of our approach in helping engineers to better accumulate lesson-learned knowledge for an enterprise. The accumulated wiki web pages in IWkS can be viewed as a large high-quality level lesson-learned knowledge repository, which stores engineers' wisdom of product design for enterprises' knowledge reuse.

This research has several limitations. The current precision and recall is still not so high that continuous improvement has to be done. The proposed approach may not work if experts change their expertise domain (or specialize in small part of the domain) or get promoted. We would like to extend our work and design larger experiments to address these issues.

Acknowledgments. The research is supported by National Nature Science Foundation of China (No. 70971085, 71271133), Science and Technology commission of Shanghai (No. 13111104500), Shanghai Municipal Education Commission (13ZZ012), Research Fund for the Doctoral Program of Higher Education of China (No. 20100073110035). The authors would like to thank all the participants for their efforts in our experiments.

References

1. Carlucci, D. and G. Schiuma: Knowledge assets value creation map: assessing knowledge assets value drivers using AHP. *Expert Systems with Applications* 32(3), 814-821 (2007)
2. Taljaard, L. and G. Smith: Mapping the relationship between knowledge management and information architecture. In: 4th International Conference on Intellectual Capital, Knowledge Management and Organisational Learning, pp. 403-410. (2007).
3. Lu, Z., J. Zuhua, and S. Hai-Tao: Distributed knowledge sharing for collaborative product development. *International Journal of Production Research* 49(10), 2959-2976 (2011)
4. Lykourantzou, I., et al.: CorpWiki: A self-regulating wiki to promote corporate collective intelligence through expert peer matching. *Information Sciences* 180(1), 18-38 (2010)
5. Bughin, J. and J. Manyika: How businesses are using Web 2.0: A McKinsey global survey. *McKinsey Quarterly Web Exclusive*. McKinsey and Company, (2007)
6. Zhao, H., W. Lu, and Ieee: Using Document Weight Combining Method for Enterprise Expert Mining. In: *International Conference on Wireless Communications, Networking and Mobile Computing, Vols 1-15*, 3721-3723 (2007)
7. Kumar, A. and N. Ahmad: ComEx miner: expert mining in virtual communities. *International Journal of Advanced Computer Science and Applications* 3(6), 54-65 (2012)
8. Yukawa, T., et al.: An expert recommendation system using concept-based relevance discernment. In: *International Conference on Tools with Artificial Intelligence*, pp.257-264. (2001)
9. Xuan, Z., et al.: An expert recommendation system of project reviewing based on the content-based model. *Journal of Information and Computational Science* 6(1), 1-7 (2009)
10. Davoodi, E., K. Kianmehr, and M. Afsharchi: A semantic social network-based expert recommender system. *Applied Intelligence* 39, 1-13 (2012).
11. Wang, G.A., et al.: ExpertRank: A topic-aware expert finding algorithm for online knowledge communities. *Decision Support Systems* 54(3), 1442-1451 (2013)
12. Porter, M.F.: An algorithm for suffix stripping. *Program: electronic library and information systems* 14(3), 130-137 (1980)
13. Salton, G. and C. Buckley, Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5), 513-523 (1988)
14. Liu, D., et al., Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing and Management* 49, 312-329 (2013)