Knowledge-Based Systems 75 (2015) 173–183

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Modeling knowledge need awareness using the problematic situations elicited from questions and answers

Bo Song, Zuhua Jiang *, Xinyu Li

Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai 200240, PR China

ARTICLE INFO

Article history:
Received 29 June 2014
Received in revised form 29 November 2014
Accepted 1 December 2014
Available online 6 December 2014

Keywords:
Knowledge need
Knowledge application context
Active knowledge system
Questions and answers
Context awareness

ABSTRACT

Being aware of people's unuttered knowledge needs is the prerequisite of providing "active" and "in time" knowledge assistance. However, such an awareness of knowledge needs has been achieved at a high cost since most existing methods rely on the manually defined rules or a large amount of user data to work. In this paper, we formulate the problematic situations in task processing as *knowledge application context* (KAC), and propose to elicit KACs semi-automatically from domain Q&A archives. Assuming that the KACs frequently occurring and semantically matching with the user's task context are more likely to imply the knowledge needs of the user, we design a mechanism of *knowledge need awareness* (KNA) to predict users' knowledge needs in complex tasks. Experimental results show that the proposed method has significantly outperformed the information retrieval approaches used as baselines. The study provides a new method for reusing the contextualized knowledge in Q&A and thereby opens up a new way to build efficient active knowledge systems.

1. Introduction

There has been a strong wish that intelligent systems could understand people's unuttered knowledge needs and provide active assistance to free people from time-consuming knowledge seeking. For this purpose, many different methods have been proposed. For example, one way to realize active knowledge assistance is to let domain experts elaborate on a chosen task, so that accurate predictions of knowledge needs can be made with the thoroughly enumerated problematic situations [1,34,47,49]. Another method starts with collecting enough samples of a task, and then uses machine learning to learn the task status triggering knowledge needs [28,39,42,46]. Recommendation techniques exploiting user ratings on knowledge items have also been used for developing active knowledge systems [2,19,37,54].

However, the aforementioned methods have not solved all the types of knowledge needs with efficiency. For example, suppose an engineer has just received an email pointing out some mistakes in his previous report. In this case knowledge need awareness (KNA) should be able to detect what difficulties the engineer may have in report revision and then automatically find the solutions of the perceived difficulties. For active knowledge systems built on expert knowledge, to detect a specific difficulty such as "how to simulate accidental load on chains with nonlinear material property," considerable efforts are required for defining the related concepts, task processes and rules. Machine learning-based methods face a similar problem with system overhead – when knowledge needs become fine-grained, the task samples used for training a task–need mapping model will be hard to acquire. The rating-based knowledge recommenders usually sustain a long-lasting user interest, which makes them not sensitive to the instant change of a user's task context. In the above example, a rating-based knowledge recommender would probably maintain the same recommendation list before and after the engineer receives the comment.

In this paper, users' knowledge needs are deemed as triggered by the complex tasks and related with problem-solving knowledge, and the knowledge needs are to be predicted without manually enumerating the enormous problematic situations. To meet this challenge, we resort to the domain Q&A archives holding plenty of problem descriptions. Based on these problem descriptions, we assess when people will meet what difficulty, and thereby get the understanding about what is the needed knowledge in the current situation.

The paper is organized as follows. In Section 2, we analyze some existing works about knowledge need awareness (KNA) and make a categorization of the conventional methods. In Section 3, the concept of knowledge application context (KAC) is defined and the framework of KAC-based KNA is proposed. Section 4 details the

* Corresponding author. Tel.: +86 021 3420 6819.
E-mail address: zhjiang@sjtu.edu.cn (Z. Jiang).

http://dx.doi.org/10.1016/j.knosys.2014.12.004
0950-7051/

implementation of the KAC-based KNA. Section 5 uses an example to show how KNA works in a typical knowledge-intensive environment, following which the comparative evaluation of different KNA methods is conducted. Conclusion and some possible improvements to be made in the future comprise the last section.

2. Knowledge need awareness

People processing knowledge-intensive tasks such as engineering design, software development and business management face frequent knowledge shortage and have become the target of many knowledge support systems [14,20,28]. Among different types of knowledge support, the "active knowledge provision" eliminates the necessity for people to enter explicit queries and automatically fetches people useful knowledge according to the perceived knowledge needs. Regarding what kinds of knowledge needs are perceived and how they are perceived, the existing KNA methods can be classified into three types: the knowledge engineering (KE) based approaches, the machine learning (ML) based approaches and the information recommenders.

2.1. KE-based approaches

KE-based KNA features the manual enumeration of various problematic situations. The KnowMore system [1] is a typical system of this kind. To provide proactive desktop task support, the KnowMore system models a business process with conventional workflow tools and enriches each knowledge intensive operation in the process with predefined query templates to be instantiated at the runtime of the process. The placement of query templates, which is the key to KnowMore, relies on the knowledge and efforts of domain experts. Staab and Schnurr [49] propose a similar task support system dealing with weakly structured business process and semi-structured documents. It uses Petri Nets to model the switches between tasks. When task switches are detected, some precompiled queries are submitted to the knowledge base to retrieve knowledge. The AKSIO project [34] establishes an active socio-technical system aimed at facilitating the knowledge transfer between drilling projects. An ontology-supported search engine is planted into the company's work process to enable the provision of cues such as in what case a drilling equipment is going to fail. CALO [47], a personal task management agent, is developed to reduce the information overload faced by a professional. CALO's ability to recognize the chance of performing a helpful action is based on a theory of user desires and a model of helpfulness, which needs expert knowledge to be materialized. The cognitive business intelligence system FACETS [33] is proposed to help people better understand their decision situations. It includes a knowledge navigation interface strengthened by automated display of task-relevant decision situations. The formalized experience, a date warehouse and an ontology are used by FACETS to retrieve relevant decision situations.

KE-based KNA are highly ad hoc due to the specialty of the application and the subjectivity of its developers. This makes it difficult to be transplanted to other applications or be mutually compared. In addition, KE-based KNA requires all task-relevant concepts be clearly defined in order to facilitate reasoning, therefore high cost on system development and maintenance is inevitable. Nevertheless, with considerable efforts put into the collecting, tiding and use of expert knowledge, KE-based KNA is capable of processing detailed knowledge needs and enjoys high accuracy.

2.2. ML-based approaches

To be aware of people's unuttered knowledge needs, knowing what they are doing is very important. Starting with desktop task recognition [44], Shen et al. conduct a series of studies [45,46] aimed at providing active support for knowledge workers. The KNA methods used in these studies differ from the KE-based approaches evidently as they require no human effort to specify the problematic situations. Instead, machine learning techniques are employed to learn the task status that triggers knowledge needs. In the latest version of Shen's work [46], a multinomial Naïve Bayes predictor is trained using the task samples collected from the Lotus Activities System [15]. Lokaiczyk et al. [28] propose a just-in-time e-learning system adopting different machine learning techniques. The system uses the desktop context as input, which includes the operating system events, user files, network stream and clipboard content. With a similar understanding of desktop context, Rath et al. [39] develop a context-aware knowledge service that incorporates SVM (support vector machine) based task classification and IR (information retrieval) based knowledge resource recommendation. Rath et al. [40] later propose a User Interaction Context Ontology to map the raw context data into a unified model. A more recent work on task recognition investigates the performance of different classifiers including KStar, decision tree, Naïve Bayes and multilayered perceptron [22]. The PASTREM system [42] uses LDA (latent Dirichlet allocation) and topic model to identify the topic of a user's activity. The unsupervised learning method has made PASTREM less dependent on annotated task samples, but the system still needs a lot of structured input which includes the attributes and content of users' accessed information items, users' time spent with each information item, the activities performed on each item as well as an ontology organizing the above information.

The main problem with ML-based KNA is that it has to observe many examples before it can recognize a knowledge need pattern. The examples may be expensive to acquire – for each task sample the user has to tell the name of the task that he/she is processing and indicate the beginning and end of the task execution. At the same time, the selection of task features for task classification becomes more and more relied on expert knowledge, which is a diversion from machine learning's original intention.

2.3. Information recommenders

Information recommenders refer to the type of systems that can actively provide information for users but cannot tell what problems they are solving. For example, the Syskill & Webert system [37] recommends a user interesting webpages by training a binary classifier from the user's positive and negative ratings on browsed webpages. For such a system, users' specific information needs are replaced by the "like" and "dislike" metric to guide the selection of information items. The information items sharing similarities with what the user is currently browsing may be worth the attention of the user. Following this idea, the so called "just-in-time information retrieval" [7–9] uses the keywords extracted from a user's activated documents to retrieve potentially useful information. The techniques used by the aforementioned information recommenders belong to the type of content-based filtering (CBF), which often suffers from the problem of over-specialization when used alone. Over-specialization means the user is limited to being recommended items similar to those already read [29]. To diversify the topics of recommended information, Li et al. [24] reckon in the user comments when making news recommendation. Language models and Kullback–Leibler (KL) divergence are used by them to measure the similarity between news items. Other than directly matching the content of knowledge, Zhen et al. [53] consider the coincidence between knowledge context and user context, where the context dimensions include time, location, role, type, domain and other information acquired from a management information system.