Journal of Biomedical Informatics 45 (2012) 557–565

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

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

Employing UMLS for generating hints in a tutoring system for medical problem-based learning

Hameedullah Kazi [a,*], Peter Haddawy [b], Siriwan Suebnukarn [c]

[a] Department of Electrical Engineering & Computer Science, Isra University, Pakistan
[b] United Nations University International Institute for Software Technology, Macao
[c] School of Dentistry, Thammasat University, Thailand

ARTICLE INFO

Article history:
Received 15 July 2011
Accepted 28 February 2012
Available online 13 March 2012

Keywords:
Ontology
Hint generation
Intelligent tutoring systems
Medical PBL
UMLS
Knowledge acquisition bottleneck

ABSTRACT

While problem-based learning has become widely popular for imparting clinical reasoning skills, the dynamics of medical PBL require close attention to a small group of students, placing a burden on medical faculty, whose time is over taxed. Intelligent tutoring systems (ITSs) offer an attractive means to increase the amount of facilitated PBL training the students receive. But typical intelligent tutoring system architectures make use of a domain model that provides a limited set of approved solutions to problems presented to students. Student solutions that do not match the approved ones, but are otherwise partially correct, receive little acknowledgement as feedback, stifling broader reasoning. Allowing students to creatively explore the space of possible solutions is exactly one of the attractive features of PBL. This paper provides an alternative to the traditional ITS architecture by using a hint generation strategy that leverages a domain ontology to provide effective feedback. The concept hierarchy and co-occurrence between concepts in the domain ontology are drawn upon to ascertain partial correctness of a solution and guide student reasoning towards a correct solution. We describe the strategy incorporated in METEOR, a tutoring system for medical PBL, wherein the widely available UMLS is deployed and represented as the domain ontology. Evaluation of expert agreement with system generated hints on a 5-point likert scale resulted in an average score of 4.44 (Spearman's $\rho = 0.80$, $p < 0.01$). Hints containing partial correctness feedback scored significantly higher than those without it (Mann Whitney, $p < 0.001$). Hints produced by a human expert received an average score of 4.2 (Spearman's $\rho = 0.80$, $p < 0.01$).

1. Introduction

Problem-based learning (PBL) has become increasingly popular in medical schools as a means of training students and equipping them with the required clinical reasoning skills. A typical PBL session in the medical domain comprises a group of 6–8 students who work in collaboration to solve a given problem scenario [1]. Paying individual attention to a small group of students can place a heavy burden on faculty time, which is very costly. This is particularly true for medical faculty, who often have limited time to dedicate to teaching. Intelligent tutoring systems offer an attractive alternative in helping to train the students with the required clinical reasoning skills at no incremental cost per student.

Intelligent tutoring systems are interactive software applications that present a problem to the students in a particular domain. The students form their solution to the problem using the tutoring system interface. The system assesses the student solution and returns appropriate hints as feedback to guide the student towards a correct solution.

Tutoring systems normally contain either a set of approved solutions or, a mechanism that generates approved solutions to the problems presented to the students. Assessment of the student solution and feedback returned is tailored to be effective only within the knowledge confines of the approved solutions. Tutoring systems are typically unable to assess the partial correctness of student solutions when they fall outside the scope of the approved ones. Furthermore, for the purpose of solution representation, students are restricted to the choice of domain concepts from a custom built repository which is often quite narrow. Such characteristics lend themselves to a tutoring approach that is fairly brittle and quite opposed to how a human tutor would behave. A human tutor allows a diverse choice of domain concepts, assesses where the student solution lies in the broad knowledge space, acknowledges the partially correct aspects of the solution and guides the students back to the correct solution. Thus in order for a tutoring system to exhibit robust tutoring, it needs a broad knowledge base to allow students to explore a large space of solutions and work

* Corresponding author. Address: Isra University, 313 Hala Road, Hyderabad, 71000 Sindh, Pakistan. Fax: +92 22 2030185.

E-mail addresses: hkazi@isra.edu.pk (H. Kazi), haddawy@iist.unu.edu (P. Haddawy), ssiriwan@tu.ac.th (S. Suebnukarn).

1532-0464/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jbi.2012.02.010

creatively, while still being able to steer them towards a correct solution if they get off track.

An ontology presents great potential for reuse and as a knowledge base that could be exploited for reasoning purposes. Several tutoring systems have employed ontologies [2–4], but they require extensive effort in encoding the relevant knowledge into the ontology. The Constraint Acquisition System [5] uses a more feasible method of encoding the ontology constraints by learning from examples, but its initial design still needs to be defined manually.

The construction of a tutoring system typically requires knowledge acquisition in the three areas of domain model, student model and pedagogical model. Acquiring and encoding the relevant knowledge can lead to a large overhead in the development time of a tutoring system [6,7]. Attempts to expand the system and reuse the existing domain model for the rapid addition of new problems or cases are often hindered by the daunting task of acquiring the student model.

While the importance of the student model has been advocated [8], the design of some tutoring systems has excluded the student model based on the needs of the tutoring task [9]. Similar to Andes [9], our system too, does not use assessment to select the next task to be offered to the student. Because of the extensive effort required, tutoring systems often excel in one or two of the three models mentioned above and maintain a more simplified form of the remaining ones [10].

The development time for a tutoring system has also come under scrutiny in the comparison between Model Tracing (MT) and Constraint Based Modeling (CBM) [11,12]. Kodaganallur et al. [11] and Mitrovic et al. [12] have acknowledged the tradeoff between the reduction in development time and the quality of hints generated. The development time required to add a case is expected to vary based on the nature of the task domain. For the domain of statistical hypothesis testing, Kodaganallur et al. [11] report the development time of 5 person-days for problem modeling and 18 person-days for encoding the relevant knowledge in the case of CBM, whereas the development time was greater for MT. CBM simplifies the creation of new cases and has a reduced development time; however, its hints are not as effective and specialized as those in MT [11,12].

In order to ease the knowledge acquisition bottleneck, Martin and Mitrovic [13] adopt a CBM approach, where the student model is an overlay of the domain model constraints. Their student model simply contains a score of the times a constraint has been satisfied or violated during problem solving. However, defining and encoding the constraints is still a burdensome task. Defining the constraints would be even a greater burden and challenge for an ill-defined domain such as medical PBL [14].

In the domain of medical PBL, students may arrive at a solution from a variety of reasoning paths [15], making it a daunting task to build the student model. Based on our previous experience with the COMET system for medical PBL [16], it takes about one person-month to build the student model for each problem scenario. Modeling the diverse set of reasoning paths would be even more challenging if the system is expected to be robust in its tutoring approach by allowing students to explore a much broader solution space as mentioned above.

We extend our work on expanding the plausible solution space [15] by deploying the widely available knowledge source, the Unified Medical Language System (UMLS) [17], as the domain ontology in the METEOR tutoring system for medical PBL. In previous work [23] we had also presented a tool for authoring medical PBL cases using UMLS. In this paper we present a strategy for alleviating the overhead required to expand the tutoring system in adding new cases by omitting the student model. We exploit the structure of the domain ontology to assess the partial correctness of student solutions and generate hints that are relevant to the student activity during problem solving [30]. Furthermore, the time and effort required to add a new problem scenario to the tutoring system is also reduced.

2. Related work

2.1. UMLS in intelligent systems

The UMLS has been used for various purposes in the biomedical informatics domain, such as terminology development, lexical matching and biomedical document understanding. Qing and Cimino [31] extract knowledge of disease–chemical relationship from the UMLS for purposes of enriching electronic patient records for online perusal.

Mendonca and Cimino [26] describe work on extracting knowledge from MEDLINE citations for purposes of building a knowledge base. They analyze the search results to determine which semantic types are relevant to what kind of questions in Evidence Based Medicine, such as diagnosis, etiology, therapy and prognosis.

Achour et al. [28] describe a knowledge acquisition tool and how it could be employed to use and share knowledge from UMLS. Their work is primarily based on providing knowledge bases for clinical decision support systems. Their focus is not to use the semantic types and concepts in UMLS for reasoning purposes, but to use UMLS knowledge sources as a repository of terms from which a domain ontology could easily be constructed.

2.2. Semantic similarity

In order to provide students with partial correctness feedback, METEOR assesses the closeness of the student solution to a correct solution explicitly encoded into the system. This closeness is measured through the semantic similarity or semantic distance between relevant concepts.

Beginning with simple path length based measures [32,33] to advanced information theoretic metrics [34,35] researchers have developed methods through which, similarity between two concepts in an ontology, could be defined in quantitative terms. Most similarity measures determine the lowest common subsumer (LCS) of the two concepts, to compute the path length from one node to the other node through this LCS. The LCS is the lowest node in the hierarchy that is a common ancestor to both the nodes, between which semantic distance is to be measured.

There has been growing interest in defining and applying measures of semantic distance, for medical terminologies and the UMLS. Caviedes et al. [27] develop a quantitative metric that can enable intelligent systems to differentiate between concepts in UMLS and measure their semantic distance. They describe their results for PAR (parent–child) links between concepts based on three terminologies within UMLS, MeSH, SNOMED-CT and ICD9CM. They adopt a simple edge counting procedure to compute the conceptual distance between two concepts over the shortest path between them, while simply mentioning the depth of the concepts in the hierarchy, as a possible influencing factor in the similarity measure.

Al-Mubaid and Nguyen [22] present an information theoretic approach to compute the semantic distance between two given concepts in an ontology. They use a cluster-based approach where the depth of the tree cluster, containing the relevant concept nodes is used along with a scaled measure of the path length between respective concept nodes. Concepts that lie deeper in the ontology tree will be more similar based on the specificity of information.

Pedersen et al. [25] discuss and analyze a set of existing semantic similarity measures and describe a context vector measure based on medical corpora. They compare the context vector measure with existing measures as applied to a commonly used dataset of