Knowledge Component Suggestion for Untagged Content in an Intelligent Tutoring System

Mario Karlovčec[1], Mariheida Córdova-Sánchez[2], and Zachary A. Pardos[3]

[1] Department of Computer Science, Jozef Stefan Institute, Slovenia
[2] Department of Computer Science, Purdue University, USA
[3] Department of Computer Science, Worcester Polytechnic Institute, USA
mario.karlovcec@ijs.si, cordovas@purdue.edu, zpardos@wpi.edu

Abstract. Tagging educational content with knowledge components (KC) is key to providing useable reports to teachers and for use by assessment algorithms to determine knowledge component mastery. With many systems using fine-grained KC models that range from dozens to hundreds of KCs, the task of tagging new content with KCs can be a laborious and time consuming one. This can often result in content being left untagged. This paper describes a system to assist content developers with the task of assigning KCs by suggesting knowledge components for their content based on the text and its similarity to other expert-labeled content already on the system. Two approaches are explored for the suggestion engine. The first is based on support vector machines text classifier. The second utilizes K-nearest neighbor algorithms employed in the Lemur search engine. Experiments show that KCs suggestions were highly accurate.

Keywords: Intelligent Tutoring Systems, Text Mining, Knowledge Components, TextGarden, Lemur, Bag-of-Words.

1 Introduction

When designing exercises within the learning software, appropriate knowledge components should be assigned to them. "A knowledge component is a description of a mental structure or process that a learner uses, alone or in combination with other knowledge components, to accomplish steps in a task or a problem." [1] The process of assigning knowledge components to the exercises can be a time consuming job, since the number of possible knowledge components can be very large. In order to help the tutor or course designer in writing exercises we have proposed two approaches that suggest knowledge components. The first approach is based on text mining [2] and SVM classification algorithm and the second is based on a search engine with a KNN classification algorithm [3]. These two approaches can be used for a system that could encourage the course designers to assign knowledge components to new exercises they design, as well as to existing exercises that do not have knowledge components assigned to them.

S.A. Cerri et al. (Eds.): ITS 2012, LNCS 7315, pp. 195–200, 2012.
© Springer-Verlag Berlin Heidelberg 2012

2 Related Work

This work continues the line of research proposed by Rose *et al.* [4] and expands on the prior art by applying a variety of optimizations as well as evaluating the algorithms on numerous KC models of varying granularity. The work by Rose et al. presented KC prediction results on a model of 39 KCs but skill models have since increased in complexity. We investigate how KC prediction accuracy scales with larger KC models and which algorithms adequately meet this challenge.

The necessity of associating knowledge components with problem solving items is shared by a number of tutoring systems including The Andes physics tutor [5], The Cognitive Tutors [6] and the ASSISTments Platform [7]. The Andes and Cognitive tutors use student modeling to determine the amount of practice each individual student needs for each KC. The student model that these tutors use is called Knowledge Tracing [6], which infers student knowledge over time from the history of student performance on items of a particular KC. This model depends on the quality of the KC model to make accurate predictions of knowledge.

The KC association with items in a tutor is typically represented in an *Item* × *KC* lookup table called a Q-matrix [8]. Methods such as Learning Factors Analysis [9] have been proposed to automate the improvement of this Q-matrix in order to improve the performance of the student model. Recently, non-negative matrix factorization methods have been applied in order to induce this Q-matrix from data [10]. The results of this work are promising but its applications so far are limited to test data where there is no learning occurring and only to datasets with only around five KCs, where these KCs represent entirely different high level topic areas such as Math and English which do not intersect. All the student modeling and Q-matrix manipulation methods have so far not tapped any information in the text of the items they are evaluating. This paper will make the contribution of looking at this source of information for making accurate KC predictions. While this paper focuses on text mined KC suggestion to aid content developers, this technique is relevant to those interested in Q-matrix improvement as well.

3 The ASSISTments Platform

The dataset we evaluated comes from The ASSISTments Platform. The ASSISTments platform is a web based tutoring system that assists students in learning, while it gives teachers assessment of their students' progress. The system started in 2004 with a focus on 8th grade mathematics, in particular helping students pass the Massachusetts state test. It has since expanded to include 6th through 12th grade math and scientific inquiry content.

A feature that sets ASSISTments apart from other systems is its robust web based content building interface [7] that is designed for rapid content development by system experts and teachers alike. Teachers are responsible for a growing majority of the content in ASSISTments. While the content has been vetted and verified as being of educational value by ASSISTments system maintainers, the content often lacks meta