Expert Systems with Applications 42 (2015) 1362–1375

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Using Datalog to provide just-in-time feedback during the construction of concept maps

Francisco J. Álvarez-Montero [a,*], Fernando Sáenz-Pérez [b], Antonio Vaquero-Sánchez [b]

[a] Facultad de Ciencias de la Educación, Universidad Autónoma de Sinaloa, C/Platón Platón 856, Villa Universidad, 80010 Culiacán, Mexico

[b] Departamento de Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid, C/Prof. José García Santesmases, s/n, E-28040 Madrid, Spain

ARTICLE INFO

Article history:
Available online 30 September 2014

Keywords:
Concept maps
Just-in-time feedback
Learning from performance errors
Binary relations
Constraints
Entity-Relationship schema
Datalog

ABSTRACT

Concept maps have been extensively used in education, especially in science teaching. There is strong evidence that their use is associated with increased knowledge transfer and retention across several instructional conditions, settings and methodological features. However, constructing a concept map is complex and difficult for students, especially newbies. Consequently, there is a necessity to provide feedback to the learners during the authoring of their concept maps. There are several concept-mapping tools that provide feedback but none of them provide immediate or just-in-time (JIT) feedback. This kind of feedback is important for two reasons: First, low achieving or low mastery students benefit greatly from this type of feedback. Second, when students start out badly, with incorrect propositions, they tend to continue with further incorrect propositions until the map is grossly incorrect and JIT feedback could prevent this situation. This paper presents a practical application of Ohlsson's theory of learning from performance errors to provide JIT feedback during the construction of concept maps. It is shown that by creating an Entity-Relationship (E-R) schema that incorporates additional elements into the standard schema for concept maps, the schema can be implemented with Datalog, benefiting from the use of its deductive features to provide immediate feedback to the learner. Finally, some field related examples are provided.

1. Introduction

Concept maps (Novak & Gowin, 1984) are the product of mapping one or more categorical propositions (Hurley, 2010). These propositions are composed of two classes, known as the referent and the relatum, and a term, representing a binary or dyadic relation (Cohen & Nagel, 1993). Graphically, these elements take the form of nodes and labeled directed arcs, respectively. The nodes represent concepts or ideas within a subject area or domain, and the labeled directed arcs are binary relations which explain how two concepts are related.

As an educational tool, concept maps are based on the notion that concept interrelatedness is an essential property of knowledge, and the empirical finding that content understanding (for example, of a school subject) is represented by well-structured knowledge (O'Neil and Klein, 1997; Ruiz-Primo, Shavelson, Li, & Schultz, 2001). They have been applied to enhance both individual and collaborative learning, and there is strong evidence that their use is associated with increased knowledge transfer and retention across several instructional conditions, settings and methodological features (Daley & Torre, 2010; Horton et al., 1993; Nesbit & Adesope, 2006). Additionally, their use in education, according to different researchers (Anohina-Naumeca, Grundspenkis, & Strautmane, 2011; Ruiz-Primo et al., 2001; Yin, Vanides, Ruiz-Primo, Ayala, & Shavelson, 2005), can be characterized along a continuum from high-directed to low-directed. The more elements are provided to the learners, the higher the degree of directedness and vice versa. Fig. 1 shows some of the components of concept maps that can be provided by the teacher or that can be left for the students to create on their own.

However, despite their graphical simplicity, and no matter the degree of directedness, the construction of concept maps is complex and difficult for students, especially for newbies (Chang, Sung, & Chen, 2001; Cimolino, Kay, & Miller, 2003). Consequently, learner support or feedback during the construction of a concept map is recommended (Coffey et al., 2003). Feedback helps learners determine performance expectations, judge their level of understanding, and become aware of misconceptions (Mason & Bruning, 2001). Without appropriate feedback, as Chang et al. (2001) point out, learners have few opportunities to reflect upon

* Corresponding author.
E-mail addresses: francisco_alvarez_montero@uas.edu.mx (F.J. Álvarez-Montero), fernan@sip.ucm.es (F. Sáenz-Pérez), arvsmcmg@telefonica.net (A. Vaquero-Sánchez).

http://dx.doi.org/10.1016/j.eswa.2014.08.055
0957-4174/

their own thinking, and this reduces the beneficial effects of constructing a concept map.

In this sense, feedback must be differentiated from the assessment or diagnosis of concept maps. The latter addresses the question of how to measure the quality of a concept map by assigning a score, after the time allocated for the concept mapping task is over (Anohina & Grundspenkis, 2009). Feedback, in turn, can be defined as any message generated in response to a learner's action (Mason & Bruning, 2001). There are several types of feedback. In this paper, feedback must be understood as immediate or just-in-time (JIT) feedback, which is a kind of feedback that is automatically given to the learner, when he or she commits an error. Immediate feedback has been successfully implemented in many Intelligent Tutoring Systems (Graesser, Conley, & Olney, in press; Nwana, 1990) and there is evidence showing that low achieving or low mastery students benefit greatly from this type of feedback (Mason & Bruning, 2001; Shute, 2008). Additionally, some researchers such as Cimolino et al. (2003) have found that when students start out badly, with incorrect propositions, they tend to continue with further incorrect propositions until the map is grossly incorrect. Just-in-time, in principle, could help prevent this situation.

There are several concept mapping tools that provide feedback to the learner (Anohina-Naumeca et al., 2011; Chang et al., 2001; Cimolino et al., 2003; Gouli, Gogoulou & Grigoriadou, 2009). However, in all these tools feedback is on demand (explicitly requested by the user) or it is delayed until the concept mapping task is finished. This lack of immediate feedback motivated the following research question: is it possible to provide just-in-time feedback to the learner during the construction of a concept map?

Following our previous work (Álvarez-Montero, Sáenz-Pérez, Vaquero, & Jacobo-García, 2012), in this paper, it is shown that by creating a conceptual schema that incorporates two sets of properties into the binary relations of concept maps, and implementing it as a Datalog schema, it is possible to provide just-in-time feedback for high-directed concept mapping tasks, where the concepts and relations have previously been defined. By using Datalog (Ceri, Gottlob, & Tanca, 1989), the two sets of properties can be represented as constraints and used to provide immediate feedback to the learner every time he or she makes a mistake, that is, every time a constraint is violated.

This approach to just-in-time feedback is based on the core ideas of the theory of learning from performance errors (Ohlsson, 1996, 2011) and the Constraint Based Modeling paradigm for Intelligent Tutoring Systems (Chrysafiadi & Virvou, 2013; Graesser et al., in press; Ohlsson & Mitrovic, 2007) which focus on what properties a good solution must have and posit that a correct solution can never violate the constraints that follow from these properties.

For implementation purposes, the Entity-Relationship (E-R) model (Chen, 1976) is used to create the conceptual schema, and the Datalog Educational System (DES) (Sáenz-Pérez, 2011) is the deductive system employed to capture the data, the constraints on these data, and provide the feedback. The rationale is that since Datalog and the E-R model are based on the relational data model

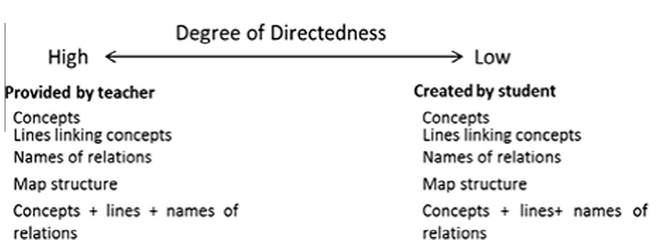

Fig. 1. Degree of directedness in concept mapping tasks.

(Chen, 1976; Ullman, 1988), the conceptual schema can be easily mapped and implemented as a Datalog schema. In addition, thanks to the more expressive data model, more complex constraints (i.e., including non-linear recursion and duplicate elimination) can be stated.

The rest of the article is organized as follows: First, Ohlsson's theory of learning from performance errors is summarized. Second, the properties of binary relations necessary to provide just-in-time feedback, their inclusion in the standard structure of concept maps and its implementation using DES are addressed and presented. Finally, some conclusions and future work are discussed.

2. Learning from performance errors

The theory of learning from performance errors (Ohlsson, 1996, 2011) states that, although humans have the innate ability to catch themselves making errors, this ability has imperfections, as Gilovich (1991) points out. Consequently, anyone can make a mistake. For instance, declaring a false statement or drawing an incorrect conclusion. The explanation is that this happens because there is a disassociation between someone's declarative and practical knowledge. Practical knowledge, also known as procedural (Ohlsson, 1996) or generative knowledge (Ohlsson & Mitrovic, 2007), is a set of rules for generating actions or behaviors that have some probability of being appropriate, correct or useful in a particular context. Declarative knowledge, in turn, enables a person to evaluate the outcome of an action or behavior, and judge it to be correct or incorrect.

Consequently, in order to learn from errors and eliminate the disassociation, a learner needs to reflect on the outcomes of his/her actions. And for that purpose, declarative knowledge in the form of integrity constraints is therefore appropriate. These constraints function as self-monitoring devices by which the learner can evaluate or judge the correctness of the action sequence generated by his/her (possibly incomplete or incorrect) practical knowledge (Ohlsson, 1996). This way, it is possible for a student to modify or update his procedural or generative knowledge base by inserting a new rule that does not violate the constraint.

Consider an example from the domain of chemistry taken from (Ohlsson, 1996, 2011). A learner is trying to construct the structural formula for an organic molecule (its so-called Lewis structure). Suppose we have a carbon atom that already has 8 valence electrons. Then the learner adds a hydrogen atom to the carbon atom and then, discovers that the carbon atom now has more than eight valence electrons. This is an error because atoms strive toward the noble gas configuration, which, in the case of carbon, requires 8 valence electrons, that is, the carbon atom already was in its noble gas configuration. The constraint that follows from this example is that: if the current number of valence electrons for a particular atom is V and the maximum number of valence electrons for atoms belonging to that substance is N, then it had better be the case that V is smaller than or equal to N (or else some error has been committed).

One can see the parallel here with the notion of integrity checking in the field of deductive databases (Colomb, 2004; Olivé, 1991), which is a process that verifies that a given base update (a set of insertions and/or deletion of base facts) satisfies a set of constraints that have the form of deductive rules, also called integrity rules. Hence, in our particular case, concept mapping can be seen as a jigsaw puzzle where a learner seeks to achieve total integrity, i.e., correctness and validity of his/her concept map, w.r.t. a prefabricated map, and receives feedback every time he/she makes an assertion that violates the constraints imposed to a relation linking a referent and a relatum. Fig. 2 shows a very general schema of the proposed notion.