Verbalization Enhanced Tutoring

Christel Kemke, Shamima Mithun

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, CANADA
{ckemke,shamima}@cs.umanitoba.ca

Abstract

Intelligent Tutoring Systems (ITS) typically contain elements of instruction, assessment, feedback and guidance for the trainee. Most of the time, the ITS is controlling the dialogue with the learner, whereas the learner has a passive, reactive role. In this paper, we suggest a technique called "verbalization", which allows users to work independently in the domain of application, and still get useful feedback and support. The principle of verbalization is to generate a natural language expression reflecting the meaning of a user's input. We developed and tested this concept in *VETS*, a Verbalization Enhanced Tutoring System for PowerLoom, a knowledge representation language based on description logic. Our studies showed that verbalization improves the persistence of users in trying to accomplish their goals and consequently leads to higher success rates in solving given tasks.

Introduction

ITS usually guide a learner through course material, assign problems to be solved, and provide feedback through error diagnosis and recommendations. More recently, ITS with natural language interfaces have been developed. These systems use natural language mainly for pre-defined feedbacks or instructions (e.g. [8]). Verbalization, on the other hand, allows a direct feedback to the user by re-formulating user inputs given as formal expressions into natural language text. Verbalization has been explored in the context of computer-generated proofs, where it improves the legibility of such proofs [1,2,3]. The hypothesis underlying our research was that ITS in formal domains like mathematics, programming or logic could become more effective by providing a verbalization of user inputs. We investigated the use of verbalization as part of a minimal tutoring system, VETS [4,4], for the knowledge representation language *PowerLoom* [7].

Description Logic and PowerLoom

Description Logic (DL) [6] is a family of knowledge representation languages based on the use of taxonomies or inheritance hierarchies to represent concepts, relations between concepts, and objects as concrete instances of concepts. For example, 'square' is a concept defined as subclass of 'rectangle'. The attribute 'side-length' relates the concept 'square' to the concept 'number'. A specific object "S1" can be created as instance of type 'square', with side-length "6". PowerLoom constructs are formal DL-like expressions to define concepts and objects, for example:

(defconcept size (?s) :<=> (member−of ?s (setof small big)))

It is difficult to learn these formal DL languages, but nevertheless PowerLoom and similar languages are often used especially by students for research and applications in knowledge representation. It thus seemed a perfect test bed for a verbalization enhanced tutoring system.

The VETS System

VETS is a passive tutoring system. Users are able to develop and query a knowledge base with the PowerLoom system as usual. The only intervention of VETS is through a feedback to the user's input.

Figure 1 provides an overview of the VETS architecture. The *Interaction Handler* serves as control and communication unit, which passes information in suitable ways between the modules. Users interact with VETS through a special dialogue window maintained by the *User Interface*. The *Domain Module* performs a syntactic analysis (parse) of the input, using a special grammar developed for the PowerLoom syntax [12]. If the Domain Module finds the user input syntactically correct, the parsed input with annotation (tags) is sent to the *Verbalization Module*. The Verbalization Module generates a natural language form of the parsed construct. Firstly, the *Discourse Planner* collects information for the verbalization of the PowerLoom construct based on the *Content Type Tree* and the contents of the knowledge base. This content information is stored with syntactic information in a feature structure (see Figure 2). The final natural language sentence is generated from this feature structure by the *Surface Realizer*, using a unification grammar. The verbalized input is displayed to the user, and VETS sends the inputted construct to the PowerLoom system for execution.

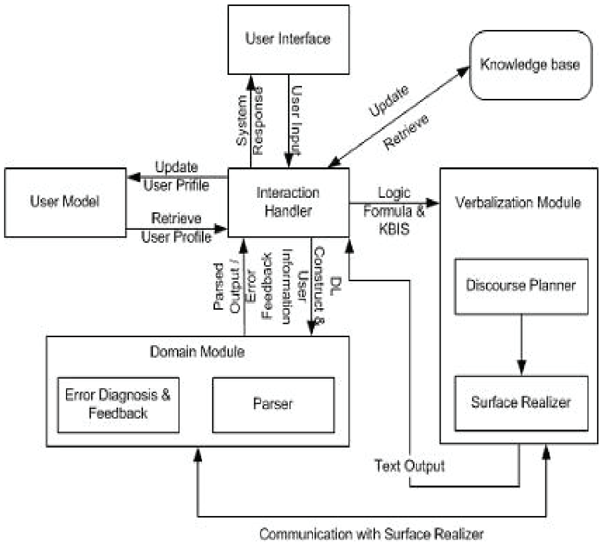

Figure 1: VETS Architecture

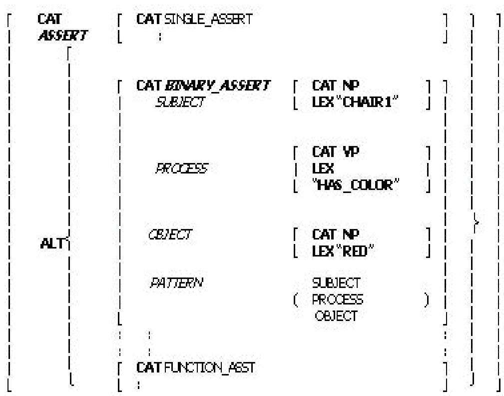

Figure 2: Feature Structure for an assert-construct.

Following are two examples of verbalization:

1. (assert (forall (?x ?y)(<=>
   (and (direction left ?x ?y)
   (and (furniture ?x)(furniture ?y)))
   (and (direction right ?y ?x)
   (and (furniture ?x)(furniture ?y))))))

   If x and y are two pieces of furniture, and x is left of y, then y is right of x.

2. (assert (left-of chair table))

   The chair is left of the table.

Evaluation, Results and Conclusion

We conducted three experimental studies, in which tasks related to building a knowledge base with PowerLoom were given to participants, typically students of computer science or related fields, after some short introduction to DL and PowerLoom. The participants of the study were split into one group working with PowerLoom plus VETS, and a control group working with PowerLoom only. The tasks were assigned to three categories, according to their level of difficulty (1=easy, 3=difficult). Measurements taken include the number of correct answers, the number of errors, the number of attempts and the time used to perform a task for each user. In addition, qualitative evaluations were obtained from the participants through questionnaires.

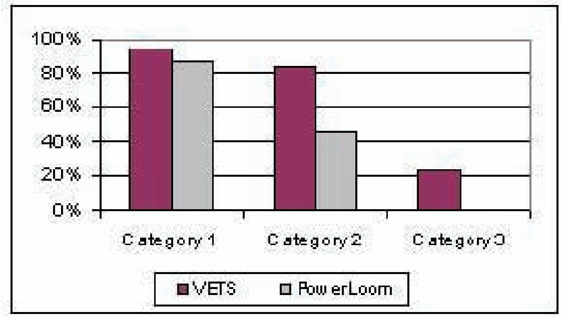

Figure 3: Comparison of success rates

The most significant and conclusive outcome of these studies is that VETS users were able to successfully complete more tasks, especially in the difficult category, than the control group (see Figure 3). In average, the number of *correct answers* or correctly solved tasks by VETS users is 1.6 times the number of correct answers given in the control group. Another noticeable outcome is that the number of *attempts* (i.e. issued PowerLoom constructs in the attempt to solve a given task) is significantly higher for VETS users than for PowerLoom-only users. From this, we can conclude that VETS users "try harder" and are seemingly more motivated to "get it right" than users of the control group, who do not have support through verbalization. Other measurements, like the time users used for the tasks, did not show a significant difference between VETS users and the control group. Subjective feedback regarding the verbalization by the participants was also mostly positive.

References

1. M. Alexoudi, C. Zinn, A. Bundy. English summaries of mathematical proofs. *Workshop on Computer-supported Mathematical Theory Development*, *IJCAR-2004*, 49–60
2. M. Holland-Minkley, R. Barzilay, R. L. Constable. Verbalization of high level formal proofs. *American Association for Artificial Intelligence*, 277–284
3. X. Huang and A. Fiedler. Proof verbalization as an application of NLG. *IJCAI-1997*, 965–970
4. C. Kemke and S. Mithun. Development of an intelligent tutor for description logics. In Proc. *Workshop on Applications of Description Logics*, Ulm, Germany.
5. S. Mithun. *Verbalization in the Context of a Tutoring System for Description Logics*, M.Sc. Thesis, University of Manitoba, 2005.
6. D. Nardi, R. J. Brachman. An Introduction to Description Logics. In: *The Description Logics Handbook*, 1–40. Cambridge University Press, 2003.
7. http://www.isi.edu/isd/LOOM/PowerLoom/
8. B. Di Eugenio, D. Fossati, D. Yu, S. Haller, M. Glass. *Natural Language Generation for Intelligent Tutoring Systems: a case study.* AIED-2005, Amsterdam