Constructional language models as building blocks for an intelligent language tutor

Whether it is for travel, work or daily communication purposes, language learning applications are becoming an increasingly popular alternative to classroom-based language courses as they allow learners to practice wherever, whenever and how often as they want. Yet, many of the existing commercial language learning software packages available today are still far removed from reaching the learning experience that a professional human tutor can offer by guiding his students through the learning material and fits the learning challenge to the level of the individual student. Commercial language learning systems such as *Rosetta Stone* instead let every student advance through the same curriculum, with an optional choice of the set of exercises that is practiced, and provide him with a minimal form of feedback that tells him if the answer is right. Such systems are similar to so-called programmed-instruction tutors that have their origin in early research in AI and education that was tightly coupled to behaviourist ideas related to reward and punishment.

Academic initiatives often offer more interesting language tutors that incorporate a full-blown Intelligent Tutoring System (ITS) architecture to allow for a more individualized task selection and feedback generation. For error diagnosis, they rely on probabilistic parsers or unification-based grammars such as HPSG to detect any minor violation of grammatical constraints (Heift & Schulze, 2007, Amaral et al. 2011). I propose a language tutoring architecture that takes this approach one step further and uses the same active language model to simulate a competent language user (domain model) and a language learner (student model), embodied in two agents that can parse and produce utterances in the target language. To do so, both agents make use of the grammar formalism *Fluid Construction Grammar* (FCG), which formalizes the constructions that each agent has and the conditions that allow these constructions to apply in parsing and production (Beuls, 2012; Steels, 2011).

In this paper, I present a constructional language tutor and demonstrate its use in the domain of Spanish verb conjugation. The tutor has the following four elements (visualized in Figure 1):

1. Learning is generally more effective when a language teacher really speaks the language that he or she is teaching at a (near) native level. Therefore, when designing an effective language tutoring system it is necessary that the artificial tutor itself is a competent speaker of the language that is being taught by the system. As a proficient language user, the tutor must thus be capable of conceptualizing and produce utterances in context or to parse and interpret them. This paper introduces the notion of a **language agent** to fulfill this need. A language agent represents an ideal speaker of a language whose linguistic skills also allow him to correct erroneous utterances of others by means of *flexibility strategies*.
2. A good teacher naturally constructs a model of his student that represents the student's skills and knowledge as a function over time. It is a kind of model that could mimic typical student utterances that are illustrative of the student's proficiency level. To operationalize such a predictive model it is convenient to reuse the three-component language agent architecture. This student model is thus implemented as a fully fledged agent, who can actively participate in the linguistic community that he finds himself in. This agent is further also

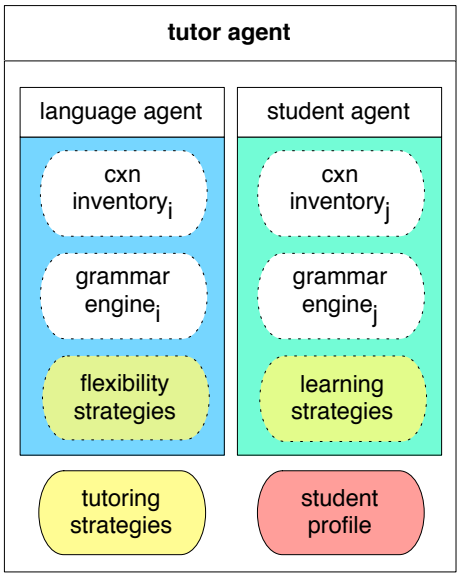

Figure 1. Apart from tutoring-specific modules, a tutor agent also has access to a fully competent language agent and a student agent that is used to predict the real student's answers.

referred to as the **student agent**. The student agent relies on *learning strategies* to expand its construction inventory and adapt the settings of its grammar engine when new situations are encountered. The linguistic knowledge of the student agent is continually aligned with the real student's utterances (Figure 2).

3. Apart from making a dynamic model of his students, a human teacher typically also applies a range of **tutoring strategies** to assist students in their problem-solving tasks. A tutoring strategy is a dynamic plan of action that stipulates future interactions with the student. To create or adapt a tutoring strategy, a teacher does not only depend on the information that is kept in the student model but he also makes use of a more general record of the student's strengths and challenges in learning.
4. Finally, a **student profile** is a record where a tutor keeps a full log of the student's activities and summaries on his progress and recent performance. This record is a personal account that belongs to a particular student and does not allow for any direct comparisons between students that are being tutored by the same tutor. The main function of the student profile is to inform the tutoring strategies component on whether to modify a strategy or add a new one.

All four components make use of a computational meta-level that runs on top of routine processing and formalizes all strategy types (flexibility, learning, tutoring) with the same architecture. The meta-level catches, inspects and repairs problems that might occur, without crashing or simply moving on to the next learning task through procedures of computational reflection (Maes, 1988).

References

Amaral, L., Meurers, D., & Ziai, R. (2011). Analyzing learner language: towards a flexible natural language processing architecture for intelligent language tutors. *Computer Assisted Language Learning*, *24*(1), 1–16.

Beuls, K. (2012). Grammatical error diagnosis in Fluid Construction Grammar: a case study in L2 Spanish verb morphology. *Computer Assisted Language Learning*, *0*(0), 1–15. doi: 10.1080/09588221.2012.724426

Heift, T., & Schulze, M. (2007). *Errors and Intelligence in Computer-Assisted Language Learning. Parsers and Pedagogues*. New York: Routledge.

Maes, P. (1988). Issues in Computational Reflection. In P. Maes & D. Nardi (Eds.), *Meta-Level Architectures and Reflection* (pp. 21–35). Amsterdam: Elsevier.

Steels, L. (2011). *Design Patterns in Fluid Construction Grammar*. (L. Steels, Ed.). Amsterdam: John Benjamins.

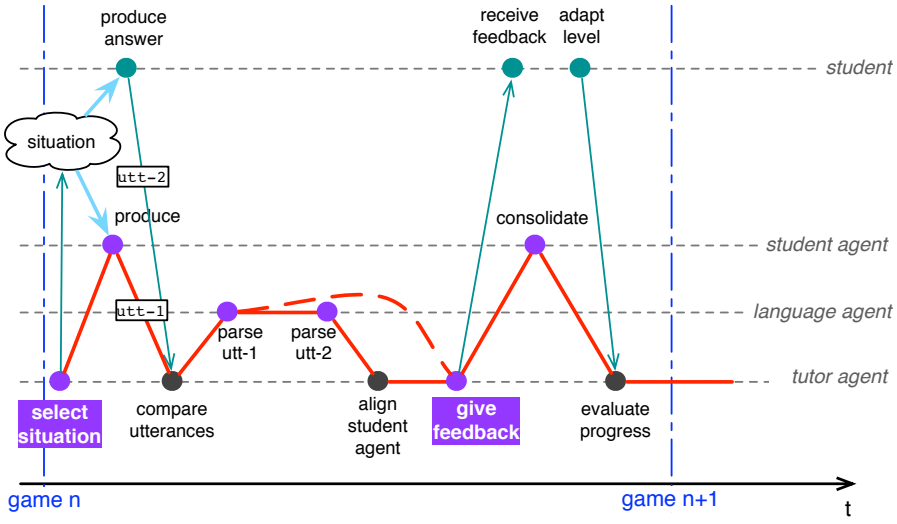

Figure 2. An example interaction between the tutor agent and the real student in a production task.