

Integrating Perceptual Learning with External World Knowledge in a Simulated Student

Nan Li, Yuandong Tian, William W. Cohen, and Kenneth R. Koedinger

School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213 USA
{nli1,yuandong,wcohen,koedinger}@cs.cmu.edu

Abstract. Systems for smart authoring of automated tutors, like SimStudent, have been mostly applied in well-defined problem-solving domains where little real-world background knowledge is needed, like math. Here we explore the generality of these methods by considering a very different task, article selection in English, where little problem-solving is done, but where complex prior perceptual skills and large amounts of background knowledge are needed. This background knowledge includes the ability to parse text and the extensive understanding of semantics of English words and phrases. We show that good performance can be obtained by coupling SimStudent with appropriate broad-coverage linguistic tools. Performance can be improved further on this task by extending one of the learning mechanisms used by SimStudent so that it will accept less-accurate production rule conditions, and prioritize learned production rules by accuracy. Experimental results show that the extended SimStudent successfully learns the tutored article selection grammar rules, and can be used to discover a student model that predicts human student behavior as well as the human-generated model.

Keywords: simulated student, English article system, learner modeling

1 Introduction

General theories and functioning simulations of how students learn have multiple uses. They can help educators to improve the understanding within domains, as well as to aid the authoring and evaluation of alternative instructional designs. To get a better understanding on how human students acquire knowledge, a lot of efforts (e.g., [2, 17, 21]) have been made to build intelligent agents that model the process of human learning in math and science.

SimStudent [17] is one such learning agent. It has been demonstrated in multiple domains such as fraction addition, equation solving, and stoichiometry [14]. Additionally, it has been shown that by integrating perceptual learning into skill learning, SimStudent can be used to find better student models than human-generated models [3]. However, most of these domains are well-defined problem-solving domains, where little real-world background knowledge is needed.

Table 1. Grammar rules in selecting appropriate articles.

Rule Name	Content	Article
generic-singular	Use “a/an” when a singular count noun is indefinite.	a/an
generic-noncount	Use “no article” with a noncount noun that is indefinite.	no article
generic-plural	Use “no article” with a plural noun that is indefinite.	no article
number-letter	Use “a/an” for single letters and numbers.	a/an
already-mentioned	Use “the” when the noun has already been mentioned.	the
same	Use “the” with the word “same”.	the

In this paper, we explore the generality of the proposed approach in a linguistic domain, article selection in English, where no complex problem solving is needed, but where complex perceptual knowledge and large amounts of background knowledge are needed. Perceptual learning in this world-knowledge rich domain requires an extensive understanding of semantics of English words and phrases and in particular, sentence parsing. There has been a long-standing interest in the natural language processing community to learn how to parse sentences correctly. Therefore, we apply one of the extensively-used linguistic tools, the Stanford parser [9], to the sentences in the problems, and integrate the perceptual representations (parse trees) of the sentences into SimStudent.

In addition, although linguistic theory has long assumed that knowledge of language is characterized by a categorical system of grammar, many previous studies have shown that language users reliably and systematically make probabilistic syntactic choices [8]. To incorporate this probabilistic aspect, we further extend SimStudent to accept less-accurate production rule conditions, and learn to prioritize learned rules by accuracy.

Experimental results show that the extended SimStudent can successfully learn how to select the correct article given a reasonable number (i.e., 60) of problems. Moreover, we use the extended SimStudent to discover human student models. The model generated by the extended SimStudent is as good as the human-generated model in predicting human student behavior.

2 English Article System

Before describing our simulated student, let us first take a look at the domain. The learning task is to acquire the English article system. There are more than 40 grammar rules to decide which article to choose.

In the current study, we took the problems from a previous study on human students [22]. There are six most-frequently used grammar rules taught in the study, as shown in Table 1. Each problem consists of one or two sentences and an empty space to be filled with an article that best completes the sentence (e.g., *Clocks measure ___ time.*). There are three choices available, *a/an*, *the* and *no article*. In the clock example, since time is uncountable, *no article* should be selected based on the rule “generic-noncount”.

Priorities exist among these six grammar rules. For example, in the problem *He drives ___ same car as he did last year*, both the condition of the rule “generic-

- | | |
|--|--|
| <ul style="list-style-type: none"> • Skill generic-noncount (e.g.,
Clocks measure ____ time.) • Perceptual information: <ul style="list-style-type: none"> • Noun the article is pointing at
(time) • Precondition: <ul style="list-style-type: none"> • Is uncountable (time) • Operator sequence: <ul style="list-style-type: none"> • Select "no article" | <ul style="list-style-type: none"> • Skill same (e.g., He drives ____
same car as he did last year.) • Perceptual information: <ul style="list-style-type: none"> • The word after the article
(same) • Precondition: <ul style="list-style-type: none"> • Is same (same) • Operator sequence: <ul style="list-style-type: none"> • Select "the" |
|--|--|

Fig. 1. The production rules “generic-noncount” and “same” in a readable format. The rule “same” has a higher priority than the skill “generic-noncount” and “generic-singular”. If the word after the article is same, “the” will be selected no matter whether the noun the article is pointing at is countable or not.

singular” and the condition of the rule “same” are satisfied, but since the rule “same” has a higher priority, the article *the* should be selected.

3 A Brief Review of SimStudent

SimStudent is an intelligent agent that inductively learns skills to solve problems from demonstrated solutions and from problem solving experience. It is a realization of programming by demonstration [13] and employs inductive logic programming [19] as one of its learning mechanisms. For more details, please refer to [17]. Recently, in order to build a more human-like intelligent agent, we have developed a model of representation learning, and integrated it into SimStudent’s skill acquisition mechanism [14].

In terms of tutoring strategy, SimStudent learns by interacting with a tutor, which can be either a human tutor or an automated tutor. Given a problem, if SimStudent does not know how to solve it, it will ask the tutor to demonstrate the next step. If SimStudent knows how to proceed, it will propose the next step, and ask for feedback from the tutor.

SimStudent learns skills as production rules. Figure 1 shows example production rules for skill “generic-noncount” and “same” in a readable format. A production rule shows “where” (i.e., perceptual information) to look for useful information, “when” (i.e., precondition) to apply the skill, and “how” (i.e., operator sequence) to proceed. For example, the rule shown in the left side of Figure 1 means given the noun that the article is pointing at (i.e., *time*), if the noun is uncountable, then select *no article*.

SimStudent has three learning components, where each of them acquires one part in the production rules. The first component is a perceptual information (i.e., “where”) learner that acquires the path to identify the useful information from its environment given a perceptual hierarchy. In our case, the environment is a graphical user interface. Each sentence is filled in a row of cells, leaving an empty cell to be filled in by SimStudent or the tutor. In the example skill “generic-noncount”, if no linguistic tool is used, the perceptual hierarchy provided to SimStudent becomes a flat list, and SimStudent may fail to learn how to identify the noun that the article is pointing at. As we will see later, with the

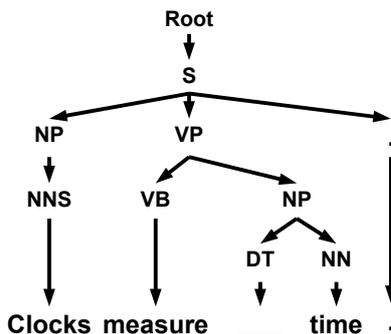


Fig. 2. The parse tree of “*Clocks measure ___ time.*” generated by the Stanford parser.

parse trees generated by the parser, SimStudent can learn the path to identify the noun. The second part of the learning mechanism is a precondition (i.e., “when”) learner, which acquires the description of desired situations in applying the skill given a set of feature predicates. The quality of the preconditions acquired largely depends on the set of feature predicates given to the precondition learner. As we will show later, SimStudent can automatically generate feature predicates based on the parse trees of the sentences. The last component is the operator sequence (i.e., “how”) learner. Given all of the demonstrated steps, the learning mechanism searches for the shortest operator sequence that could explain all of the records, using iterative-deepening depth-first search.

As we can see, the prior knowledge given to SimStudent (e.g., the perceptual hierarchy, the feature predicates, operator function) affects the learning effectiveness of SimStudent. Moreover, we want this prior knowledge to be acquired rather than programmed, since the more knowledge engineering needed, the less human-like SimStudent is. Previous studies [6] have shown that one of the key differences between experts and novices is their different representations of the world. Therefore, we have extended SimStudent to support representation learning, and integrated it into skill learning [14]. By integrating representation learning and skill learning, we can learn a tree-structured representation of the problem, automatically generate feature predicates based on the representation [15], and reduce the need of domain-specific operator functions. The representation learning mechanism used is an extended version of a probabilistic context-free grammar (pCFG) learner. For more details, please refer to [14].

4 Perceptual Learning with External World Knowledge

In spite of the promising results we have shown, the domains we have tested so far are all well-defined domains (e.g., fraction addition, equation solving, stoichiometry), where the perceptual representation captured by a pCFG can be learned without large amounts of external world background knowledge. On the other hand, article selection in English is quite different, as complex prior perceptual knowledge as well as large amounts of world knowledge is needed.

Therefore, we use an existing linguistic tool, the Stanford parser, to automatically generate the parse structure of the input sentence for SimStudent. The parse tree for the clock example is shown in Figure 2. We give these parse trees to SimStudent as the perceptual hierarchies. Based on these hierarchies, SimStudent learned that the noun that the article is pointing to is the last sibling of the article in the subtree. In the example, the non-terminal node *NP* has two children, hence, the word *time* is the noun that the article is pointing at.

Moreover, SimStudent automatically generated a set of feature predicates based on the parse tree. For example, in the parse tree shown in Figure 2, each non-terminal symbol (e.g., *NN*) is associated with a feature predicate (e.g., *(is-NN ?val0 ?val1)*). Given the parse tree, *(is-NN time Clocks-measure-time)* returns true. Topological based feature predicates such as (e.g., *(is-child-of ?val0 ?val1 ?val2)*) can also be generated, but were not used in article selection.

Lastly, we use Wiktionary¹, which is a collaborative project for creating a free lexical database in every language, complete with meanings, etymologies, and pronunciations, to generate two feature predicates (i.e., *(is-countable ?val)*, *(is-uncountable ?val)*) that evaluate whether a noun is countable or not. Note that since one word may have multiple senses, it can be both countable and uncountable at the same time.

5 SimStudent with Probabilistic Conflict Resolution

As mentioned above, although grammar rules are often modeled as a categorical system, previous studies have shown that people systematically make probabilistic choices [8]. To incorporate this feature, we developed two conflict resolution strategies that prioritize rules based on accuracy. SimStudent associates each production rule with a utility. When multiple production rules are applicable, the production rule with the highest utility is applied first.

To implement the conflict resolution strategy, we lowered the accuracy requirement of the preconditions learned by FOIL, so that preconditions that are less accurate are also included in the production rule. This modification allows SimStudent to learn more general production rules. Therefore, there are more situations where more than one production rules are applicable. However, some of them may be incorrect.

Next, SimStudent computes the utility associated with each production rule based on the correctness of the rule’s application history. We designed two ways of computing the utility. The first approach is developed based on ACT-R’s conflict resolution strategy [4], where the utility associated with production rule *i*, U_i , is calculated based on the following equation.

$$U_i = P_i G - C_i,$$

where, P_i is the probability of success of the production rule *i*, C_i is the average cost of the production rule, and G is a goal value. Please refer to [4] for details.

¹ <http://www.wiktionary.org/>

In the above approach, P_i considers all successful applications are equally important. One interesting question to ask is whether the importance of the rule application result decays as time passes. Hence, in the second approach, instead of directly computing the probability of success, SimStudent weighs recent successes more than the past ones. Each time a rule is applied correctly, it is given a constant reward, R , and the utilities of all other rules decay by another constant, d . In case of an incorrect application, the same constant value, R , is removed from the utility function. Therefore, the utility of production rule i at time t , $U_{i,t}$, is calculated by

$$\begin{aligned} U_{i,t} &= D_{i,t}G - C_i, \\ D_{i,0} &= 0, \\ D_{i,t+1} &= (-1)^{failure} R + dD_{i,t}, \end{aligned}$$

where $D_{i,t}$ is the decayed success rate at time t , *failure* is an integer that equals to 1 if the rule application is incorrect, and 0 if correct, R is the reward/punishment given to the production rule, and d is the rate of decaying.

6 Experimental Results

To evaluate the effectiveness of the proposed approach, we carried out two experiments to test, 1) whether the extended SimStudent can learn the six grammar rules; and 2) whether the extended SimStudent can predict human student behavior just as well as human-generated models.

6.1 Experimental Design

We used data collected from Wylie et. al’s [22] recent study on second language learning. The study was conducted at the University of Pittsburgh’s English Language Institute. Students ($N=99$) were adult English language learners (*mean age* = 27.9, *SD*=6.6) and participated as part of their regular grammar class. Data collection was completed within one 50-minute class period. Pre- and post-test items were identical in the form of the practice problems that students had seen during tutoring without feedback and hints. All of the student behaviors were recorded during the process, and encoded with rules applied to the problems and whether students answers are correct.

SimStudent was taught by an automated tutor that simulates the tutor used by human students, and was trained on the same 60 problems that were provided to human students. The production rules acquired were evaluated on 12 problems given to human students as test problems.

6.2 Speed of Learning

We evaluated four versions of SimStudent, 1) the original SimStudent without external world knowledge and the new conflict resolution strategy², 2) the ex-

² The conflict resolution strategy of the original SimStudent is to fire the most recently activated non-buggy production rule.

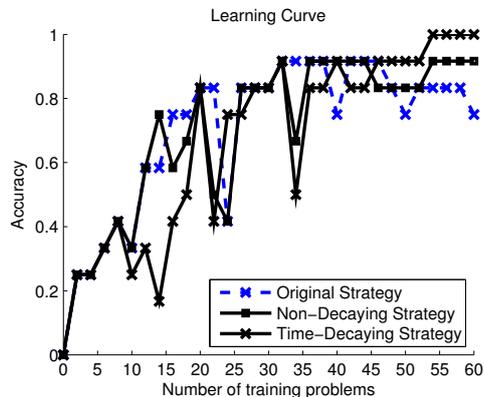


Fig. 3. Learning curves of SimStudents in article selection.

tended SimStudent with external world knowledge using the original conflict resolution strategy, 3) the extended SimStudent with external world knowledge using the non-decaying conflict resolution strategy, 4) the extended SimStudent with external world knowledge using the time-decaying conflict resolution strategy. In order to rule out the effect of other parameters, we set G and C_i to be the same across all production rules, so that the production rule priorities are decided by P_i and $D_{i,t}$. We report the average accuracy of SimStudent’s first attempts at each step over 12 test problems.

Since the original SimStudent without external world knowledge considered that all words in the sentence form a flat hierarchy, it failed to learn how to identify the noun that the article is pointing at. In fact, it learned overly-general production rules, and could not finish training in a reasonable amount of time. Therefore, the learning curve of the original SimStudent is not reported here, and should be much flatter than the extended one.

As we can see in Figure 3, all three SimStudents learn reasonably well, reaching accuracies of more than 0.75 given 60 problems. This result indicates that by integrating perceptual learning with external world knowledge, the extended SimStudent is able to successfully learn the six grammar rules. Among the three SimStudents, the extended SimStudents using the proposed conflict resolution strategies are better than the SimStudent using the original strategy. A careful inspection of the data showed that although all SimStudents learned the rule “generic-plural” and the rule “already-mentioned”, the SimStudent with the original conflict resolution strategy failed to learn that the rule “already-mentioned” is preferred over the rule “generic-plural”. For example, when given the problem, *Some planes appeared, and then ___ planes landed in a field*, the SimStudent with the original conflict resolution strategy decided to apply the “generic-plural” rule, and selected *no article*. This suggests better conflict resolution strategies can further improve SimStudent’s learning effectiveness.

The extended SimStudent using the time-decaying conflict resolution strategy learns the fastest. It reaches an accuracy of 1.00 given 60 training problems. The extended SimStudent using the non-decaying conflict resolution strategy is slightly worse than the one using the time-decaying strategy.

6.3 Fit to Human Student Data

The second experiment is to test whether the extended SimStudent can be used to discover models of human students. A student model is a set of *knowledge components* (*KC*) encoded in intelligent tutors to model how students solve problems. Applying the approach described in [3], we use SimStudent to automatically generate a student model. Each production rule or each disjunction in a rule corresponds to one KC. We compare the SimStudent-generated model with the best human-generated model constructed by domain experts. To evaluate how well the student model fits with human data, we used the Additive Factor Model (AFM) [5] to validate the coded steps. AFM is an instance of logistic regression that predicts the probability of a student making an error on the next step given each student, each KC, and the KC by opportunity interaction as independent variables. We use Akaike information criterion (AIC) and a 10-fold cross validation (CV) to test how well the generated model predicts the correctness of human student behavior without overfitting.

SimStudent successfully recovers the KCs associated with the six grammar rules. Moreover, it splits the rule “number-letter” into two KCs, one for “number” and one for “letter”. The SimStudent-generated model is **as good as** the human-generated model both in terms of AIC (6221.39 vs. 6221.49) and the root mean-squared error in cross validation (0.3769 vs. 0.3777). This suggests that SimStudent finds as good a student model as the human-generated one. Moreover, we have carried out an in-depth study using Focused Benefits Investigation (FBI) [10] to better understand the difference between the two models. Results show that among the 19 KCs in the human-generated model, 15 of them are improved, in terms of RMSE, in the SimStudent-generated model.

7 Related Work

In this paper, we extend perceptual learning with external world knowledge in a simulated student. Previous work on article selection (e.g., [22]) has shown that learning in this domain contains challenges that cause some effective instructional strategies (e.g., self-explanation) in math and science to become less effective. **Recent efforts such as the Fawltly tutor [11] have attempted to teach correct article usage by building an intelligent tutoring system.** To better understand the cause of this phenomenon and to better teach students, we constructed a learning agent that models knowledge acquisition for article selection. This required extending our model of perceptual learning with external world knowledge, and integrating it into a simulated student. Other research on ill-defined domains [16] is also related to our work, but focuses on other learning tasks.

There have been recent efforts (e.g., [2, 17, 21]) in developing intelligent agents that model student learning, but most of the existing works have been done in well-defined domains, where little real-world knowledge is needed. There has also been considerable research on learning within agent architectures [12, 1, 20], **and other efforts to incorporate machine learning to aid intelligent tutoring system authoring [18].** Unlike those theories, SimStudent puts more emphasis

on knowledge-level learning (cf., [7]) than speedup learning. Moreover, to the best of our knowledge, none of them have focused on integrating representation learning with skill learning as we have done with SimStudent.

8 Conclusion

In future work, in addition to predicting the probability of success of human students, we would also like to see what causes human students to make certain types of errors by manipulating SimStudent’s prior representation knowledge. Furthermore, **in this study, we explored the six most frequently used grammar rules in article selection. There are many other cases that are not covered by these six rules. In future studies, we plan to explore other less frequently used grammar rules in this domain** . Finally, we would like to carry out controlled simulation studies in article selection to get a better understanding of why self-explanation is no more effective than simple practice in this domain.

Constructing an intelligent agent that simulates human-level learning is an essential task in education. Previous effort has shown that by integrating a representation learning algorithm into an intelligent agent, SimStudent, as an extension of the perception module, the extended SimStudent is able to achieve comparable performance without requiring any domain-specific operator function as input in well-defined domains. In this paper, we further evaluated the generality of the approach in a world-knowledge rich domain, we extended representation learning with external world knowledge, and integrated it into SimStudent. Results show that given a reasonable number (e.g., 60) of training examples, the extended SimStudent successfully learns six frequently-used article selection rules, and can be used to find student models that predict human student behavior as well as a human-generated model.

Acknowledgements

We thank Ruth Wylie for helpful discussion, and the National Science Foundation (#SBE-0354420) for funding of the Pittsburgh Science of Learning Center.

References

1. Anderson, J.R.: Rules of the Mind. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1993)
2. Anzai, Y., Simon, H.A.: The theory of learning by doing. *Psychological Review* 86(2), 124–140 (1979)
3. authors: Details suppressed (2011)
4. Belavkin, R.V., Ritter, F.E.: OPTIMIST: A New Conflict Resoution Algorithm for ACTR. In: Proceedings of the sixth International Conference on Cognitive Modeling. pp. 40–45. Pittsburgh, PA (2004)
5. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis - a general method for cognitive model evaluation and improvement. In: Proceedings of the 8th International Conference on Intelligent Tutoring Systems. pp. 164–175 (2006)

6. Chase, W.G., Simon, H.A.: Perception in chess. *Cognitive Psychology* 4(1), 55–81 (Jan 1973)
7. Dietterich, T.G.: Learning at the knowledge level. *Machine Learning* 1(3), 287–315 (1986)
8. Hay, J., Bresnan, J.: Spoken syntax: The phonetics of giving a hand in new zealand english. In: *The Linguistic Review: Special Issue on Exemplar-Based Models in Linguistics*. pp. 321–349 (2006)
9. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. pp. 423–430. Stroudsburg, PA, USA (2003)
10. Koedinger, K.R., McLaughlin, E.A., Stamper, J.C.: Automated student model improvement. In: *Proceedings of the 5th International Conference on Educational Data Mining*. pp. 17–24. Chania, Greece (2012)
11. Kurup, M., Greer, J.E., McCalla, G.I.: The faulty article tutor. In: *Proceedings of the Second International Conference on Intelligent Tutoring Systems*. pp. 84–91. ITS '92, Springer-Verlag, London, UK (1992)
12. Laird, J.E., Rosenbloom, P.S., Newell, A.: Chunking in soar: The anatomy of a general learning mechanism. *Machine Learning* 1, 11–46 (1986)
13. Lau, T., Weld, D.S.: Programming by demonstration: An inductive learning formulation. In: *Proceedings of the 1999 international conference on intelligence user interfaces*. pp. 145–152 (1998)
14. Li, N., Cohen, W.W., Koedinger, K.R.: Efficient cross-domain learning of complex skills. In: *Proceedings of the Eleventh International Conference on Intelligent Tutoring Systems*. pp. 493–498. Berlin (2012)
15. Li, N., Schreiber, A., Cohen, W.W., Koedinger, K.R.: Creating features from a learned grammar in a simulated student. In: *Proceedings of the 20th European Conference on Artificial Intelligence* (2012)
16. Lynch, C., Ashley, K.D., Pinkwart, N., Alevan, V.: Concepts, structures, and goals: Redefining ill-definedness. *International Journal Artificial Intelligence in Education* 19(3), 253–266 (Aug 2009)
17. Matsuda, N., Lee, A., Cohen, W.W., Koedinger, K.R.: A computational model of how learner errors arise from weak prior knowledge. In: *Proceedings of Conference of the Cognitive Science Society* (2009)
18. Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., Mcguigan, N.: Aspire: An authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education* 19(2), 155–188 (Apr 2009)
19. Muggleton, S., de Raedt, L.: Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19, 629–679 (1994)
20. Taatgen, N.A., Lee, F.J.: Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors* 45(1), 61–75 (2003)
21. Vanlehn, K., Ohlsson, S., Nason, R.: Applications of simulated students: an exploration. *Journal of Artificial Intelligence in Education* 5, 135–175 (February 1994)
22. Wylie, R., Koedinger, K., Mitamura, T.: Analogies, explanations, and practice: examining how task types affect second language grammar learning. In: *Proceedings of the 10th international conference on Intelligent Tutoring Systems*. pp. 214–223. Berlin, Heidelberg (2010)