Learning Logical Reasoning Using an Intelligent Tutoring System: A Hybrid Approach to Student Modeling

Roger Nkambou[1], Janie Brisson[1], Ange Tato[1,2], Serge Robert[1]
[1]Université du Québec à Montréal, Centre de Recherche en Intelligence Artificielle
201 President-Kennedy Avenue, Montreal, Quebec, H2X 3Y7, Canada
[2]École de Technologie Supérieure
1111 Notre-Dame W. st., Montréal, Quebec, H3C 6M8, Canada
{nkambou.roger, brisson.janie, robert.serge}@uqam.ca, ange-adrienne.nyamen-tato@etsmtl.ca

Abstract

In our previous works, we presented Logic-Muse as an Intelligent Tutoring System that helps learners improve logical reasoning skills in multiple contexts. Logic-Muse components were validated and argued by experts throughout the designing process (ITS researchers, logicians, and reasoning psychologists). A catalog of reasoning errors (syntactic and semantic) has been established, in addition to an explicit representation of semantic knowledge and the structures and metastructures underlying conditional reasoning. A Bayesian network with expert validation has been developed and used in a Bayesian Knowledge Tracing (BKT) process that allows the inference of the learner skills. This paper presents an evaluation of the learner-model components in Logic-Muse (a bayesian learner model). We conducted a study and collected data from nearly 300 students who processed 48 reasoning activities. These data were used to develop a psychometric model for initializing the learner's model and validating the structure of the initial Bayesian network. We have also developed a neural architecture on which a model was trained to support a deep knowledge tracing (DKT) process. The proposed neural architecture improves the initial version of DKT by allowing the integration of expert knowledge (through the Bayesian Expert Validation Network) and allowing better generalization of knowledge with few samples. The results show a significant improvement in the predictive power of the learner model. The analysis of the results of the psychometric model also illustrates an excellent potential for improving the Bayesian network's structure and the learner model's initialization process.

Introduction

Decades of research in cognitive science show that human reasoning does not function accordingly to the rules of formal logic (e.g., (Cummins et al. 1991; Gilovich, Griffin, and Kahneman 2002; Markovits and Vachon 1989; Thompson 1994)). When looking for solutions to improve human skills in this area, several questions arise: what is important in the assessment of logical competence? What are the phenomena involved in the acquisition of logical reasoning skills? What should be the characteristics of an Intelligent Tutoring System (ITS (Nkambou, Mizoguchi, and Bourdeau 2010)) designed to support this learning? These questions can only be answered with a proper understanding of human reasoning processes and the active participation of relevant experts, including logicians, psychologists, education professionals, and IT specialists. By bringing together specialists from these different fields, the Logic-Muse project aims to study the basics of logical reasoning skills acquisition, understand the difficulties associated with this learning, and create an ITS that can detect, diagnose, and correct reasoning errors in various situations. Logic-Muse's architecture looks like a classical ITS with its three standard model components: the knowledge domain expert, the learner, and the tutor. The participatory approach adopted led to the development of essential components for constructing the Logic-Muse ITS. These components include the catalog of errors, a bayesian network for logical reasoning skills, and some remedial strategies that promote learning. The experts have previously elicited, validated, and argued these components. Logic-Muse, in its current version, implements these components for propositional logic and offers a panoply of activities allowing a learner to develop his logical reasoning skills in several situations established by the domain experts.

This paper aims to evaluate the learner model of the Logic-Muse ITS. More precisely, it uses machine learning techniques on human participants' problem-solving data to initialize and validate the ITS learner model.

The Learner Model in Logic-Muse

Logic-Muse's learner model has several dimensions. The episodic memory keeps track of the exercises performed by the learner as well as all related performance elements. The cognitive model is a Bayesian network (BN) whose nodes are the 96 units of knowledge related to reasoning, as identified by the knowledge domain specialists. The prior probabilities as well as the causal relationships between the different units of knowledge were also established beforehand by the experts. It is worth noting that causality in this case materializes the influence of one unit of knowledge in the learning of the other. The nodes are directly connected to the reasoning activities (items corresponding to Bayesian network leaves). The skills involved in the Bayesian network include the inhibition of exceptions to the premises, the generation of counterexamples to the conclusion, and the ability to manage all the relevant models for familiar, counterfactual, and abstract situations (Markovits 2013). The system's

	Familiar			CF			Abstract		
	I	G	M	I	G	M	I	G	M
MPP_FFD	1	0	0	0	0	0	0	0	0
MPP_FMD	1	0	0	0	0	0	0	0	0
MTT_FFD	1	0	1	0	0	0	0	0	0
MTT-FMD	1	0	1	0	0	0	0	0	0
AC_FMA	0	1	1	0	0	0	0	0	0
AC_FFA	0	1	1	0	0	0	0	0	0
DA_FMA	0	1	1	0	0	0	0	0	0
DA_FFA	0	1	1	0	0	0	0	0	0
MPP_CCF	1	0	0	1	0	0	0	0	0
MTT_CCF	1	0	1	1	0	1	0	0	0
AC_CCF	0	1	1	0	1	1	0	0	0
DA_CCF	0	1	1	0	1	1	0	0	0
MPP_A	1	0	0	1	0	0	1	0	0
MTT_A	1	0	1	1	0	1	1	0	1
AC_A	0	1	1	0	1	1	0	1	1
DA_A	0	1	1	0	1	1	0	1	1
I=Inhibit,G=Generate,M=Manage,CF=Contrary to fact									

Table 1: Q-matrix for conditional reasoning

estimate of student skill acquisition is continually updated every time a student responds to a problem, and that answer is used as evidence by the system to re-compute the probability that this student knew the skill before the answer. The tutor then chooses exercises based on these probabilities. The episodic memory keeps track of all the exercises performed by the learner. The Q-Matrix (built by the experts) illustrating the relationship between item types (16) and reasoning skills (9) relationship is shown in Table 1. Figure 1 shows an excerpt of the Bayesian network structure, which highlights the relationship between the items (48 grouped into the 16 item-types) and the nine (9) knowledge units (or reasoning skills) that the experts have identified.

Item Bank for Conditional Reasoning

The items used for conditional reasoning in Logic-Muse are the four logical forms of conditional reasoning: the Modus Ponendo Ponens (MPP), the Modus Tollendo Tollens (MTT), the Affirmation of the Consequent (AC), and the Denial of the Antecedent (DA).

The reason why we started with conditional reasoning is that many experimental studies in psychology have highlighted the significant variability of human reasoning with inferences of the same logical form but differing in content (e.g., (Cummins et al. 1991; Markovits and Vachon 1989; Thompson 1994), and a large part of the literature on human reasoning aims to explain such variability. Much focus has been placed on the effects of content on conditional (if-then) reasoning. According to the logical definition of the conditional connective, two of these inferences are valid, and two are invalid. The Modus Ponendo Ponens inference ("If P then Q, P is true, therefore Q is true"), referred to as MPP, and the Modus Tollendo Tollens inference ("If P then Q, Q is false, therefore P is false") referred to as MTT are both valid and lead to necessary conclusions. By contrast, the Affirmation of the Consequent inference ("If P then Q, Q is true, therefore P is true") is referred to as AC, and the Denial of the Antecedent inference ("If P then Q, P is false, therefore Q is false") referred to as DA) are both invalids since their putative conclusion does not necessarily follow from the premises. These four logical forms are declined in 3 levels of content based on a developmental model of conditional reasoning. Markovits (2013) suggests that the more a premise has familiar content, the more rapid and easy the retrieval of a counter-example to invalid conclusions will be. Consequently, the more familiar a premise is, the more reasoners will avoid reasoning errors. In this sense, Markovits has developed a graduated scale of content levels for a premise that grows in difficulty and level of abstraction. The transition between two levels of content is gradual and done in stages: when a level is mastered, the skill in the next level begins.

The familiar level contains meaningful premises representing a plausible rule linking two known entities or categories. Content effects with these types of inferences have been heavily documented. Investigations of content-related variability can fall under a general perspective called the "semantic memory framework" (De Neys, Schaeken, and D'ydewalle 2002), where the retrieval of stored knowledge primarily impacts reasoning with meaningful premises. The impact of information retrieval on conditional reasoning has been mostly observed through the effect of potential counterexamples on a putative conclusion. For the AC and DA inferences, such counterexamples are alternative antecedents, i.e., antecedents that differ from P but imply the consequent Q. For the MPP and MTT inferences, counterexamples are disabling conditions, i.e. a condition that prevents the antecedent P from implying the consequent Q. Many studies have shown that the number of potential counterexamples (Cummins et al. 1991; Thompson 1994) or the strength of association between them and the premise (De Neys, Schaeken, and D'Ydewalle 2003) determines approval rate of the four forms of conditional inference. For example, with the premise "If a rock is thrown at a window, then the window will break," reasoners will tend to accept the AC inference (a window is broken; therefore, a rock was thrown at it) less often than with the premise "If a finger is cut, then it will bleed" (a finger bleeds, therefore it has been cut). The reason is that the former premise contains many alternative antecedents, like throwing a chair, a car accident, a tropical storm, etc., that are counterexamples to the putative conclusion, while the latter contains fewer of such antecedents (a finger is crushed, etc.).

The counterfactual level contains a reversed causal rule known to be false. It allows the generation of an unrealistic category of alternative antecedents. For example, with the counterfactual premise "If a feather is thrown at a window, then the window will break", one could generate a counterfactual alternative antecedent like "throwing tissue on a window" or a disabling condition like "The window was strong enough to stay intact."

The abstract level contains if-then rules linking made-up words, e.g., "If one blops, then one will become plede." This level requires an abstract representation of the premises

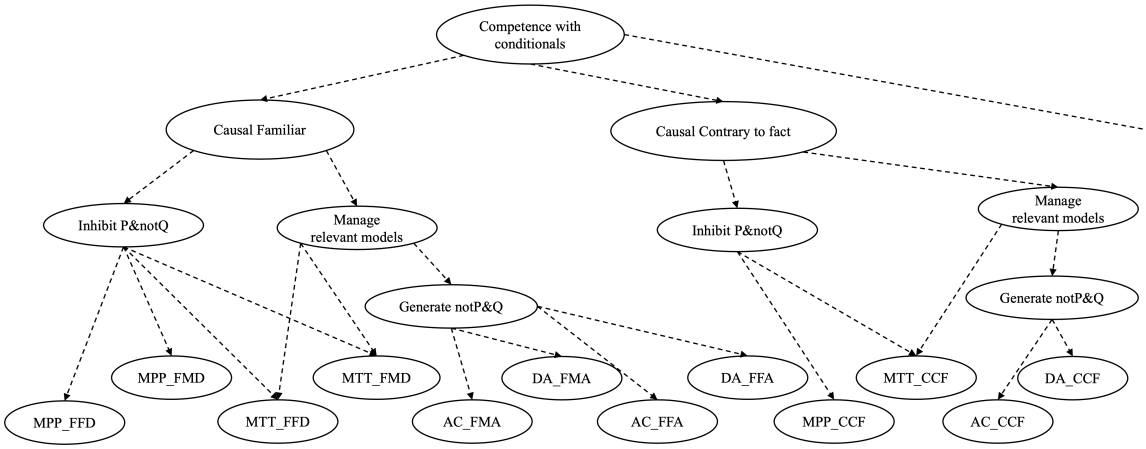

Figure 1: Excerpt of the Bayesian network structure for conditional reasoning

that can generate abstract alternative antecedents. According to (Markovits 2013), when a reasoner reaches this level, he has a complete understanding of the implicative link: he understands that for a conditional premise (unknown or abstract), an alternative antecedent can be generated regardless of background knowledge. We thus have a total of 16 item classes (item nodes in the BN).

Training a CDM Model to Validate the BN and to Initialize the Learner Model

Logic-Muse's learner model aims to represent user knowledge as accurately as possible using the Bayesian network we created. It allows for diagnosis and modeling of the learner's current state of mastery for each identified skill. Validity and reliability are thus very important features of this model. In this paper, we will provide data to initiate and evaluate our model's validity. To this end, we will use the R library CDM (George et al. 2016; Ravand and Robitzsch 2015) to run a Cognitive Diagnosis Model on input data previously collected. This CDM model is built using the item bank, a Q-Matrix (items/skills), and data from all student responses to items (De La Torre 2009). The Q-Matrix (Table 1) connects item categories to the involved skills. The resulting model is part of the learner model and allows for initial predictions of the learner's state of mastery, given his response vector.

Data Collection

Participants and procedures. A total of 294 participants were recruited online via the Prolific Academic platform.

Materials. For each of the 16 items classes, three items were constructed. This was done in order to obtain a more reliable measure for each competence class. At the beginning of the questionnaire, participants were given some instructions, as shown in Table 2.

These instructions are followed by the presentation of the 48 items, similar to the example above.

Instructions
In the following pages, we will present rules that you must assume to be true. Following the rule, an observation will be presented to you. Your task is to select the conclusion that logically follows from the information provided. Here is an example of a question that you will have to answer: Suppose it is true that: If we touch something that is very hot, then we burn ourselves. Observation: Marco touched something that is very hot. One can conclude that: • *Marco burned himself.* • *Marco did not burn himself.* • *One cannot conclude whether Marco burned himself or not.* Please indicate below that you have read the instructions. Then press the "Next" button at the bottom right to start the questionnaire.

Table 2: Instructions to the learners

Data Preparation

For each of the 48 items, participants had to choose between 3 answers (the valid one, the invalid typical one, and the invalid atypical one). Answers were encoded as "1" for valid and "0 otherwise. We then had to choose between three possible response matrices according to the number of valid responses for the three repeated measures for each of the 16 categories. Each category is encoded as 1 if the participant was successful for at least 2 out of 3 items. This particular threshold was chosen out of consistency with previous modeling choices: a majority of a successful response is the criteria to activate a competence node in our Bayesian Network. Table 3 shows some examples of items.

CDM Model Type Selection and Training

We then trained a CDM model on the 294 response patterns, which allowed us to estimate various parameters: posterior

Familiar content (F)		
Few disablers (FD), Many alternatives (MA)		
Major premise: If one jumps in a pool, then one is wet.		
MPP_FFD	Observation	Bob jumped in a pool.
	Conclusion	Bob is wet.
MTT_FFD	Observation	Paul it not wet.
	Conclusion	Paul didn't jump in a pool.
AC_FMA	Observation	John is wet.
	Conclusion	One cannot conclude
DA_FMA	Observation	Mark didn't jump in a pool.
	Conclusion	One cannot conclude.
Many disablers (Md), few alternatives (Fa)		
Major: If a match is scraped, then it will light up.		
MPP_FMd	Observation	A match was scraped.
	Conclusion	The match will light up.
MTT_FMd	Observation	The match is not lit.
	Conclusion	The match was not scraped.
AC_FFa	Observation	The match is lit.
	Conclusion	One cannot conclude.
DA_FFa	Observation	The match was not scraped.
	Conclusion	One cannot conclude.
Abstract content (A)		
Major: If one blops, then one will become plede.		
MPP_A	Observation	Mary-Ann bloped.
	Conclusion	Mary-Ann will become plede.
MTT_A	Observation	Frank did not become plede.
	Conclusion	Frank did not blop.
AC_A	Observation	Michelle became plede.
	Conclusion	One cannot conclude.
DA_A	Observation	Peter did not blop.
	Conclusion	One cannot conclude.

Table 3: Sample items in the database. Conclusion means valid conclusion.

probabilities, the goodness of fit indicator, guess (the probability that a learner could correctly answer an exercise without having the necessary skills), slip (the probability of a wrong solution while the learner had the necessary skills), tetrachoric correlations, and marginal skill probabilities. We had to choose between a Deterministic Input, Noisy "Or" gate (DINO) and a Deterministic Input, Noisy "And" gate (DINA) CDM model. We opted for the DINA model since it makes the same assumption we made in our modeling of skills: the learner must have mastered all the related skills to succeed in an item. The DINA model assumes that at least one of the related skills needs to be mastered to succeed in an item. It was thus deemed inappropriate for our study.

BN Initialization through the CDM Model

The CDM allows us to predict a user's probability of mastering the overall competence (root node) via its pretest results. For this, we use the a posteriori probabilities obtained. To do so, from a learner's vector of competence, we seek the line of the a posteriori matrix containing the same or similar vector. The joint probability matching this pattern, calculated based on the probabilities associated with each skill, is then used as the a priori likelihood (prior probability) of mastering the root node.

The first matrix indicated that the most probable vector was the "111 000 000", which means a learner masters the three competences for the familiar level of content, but no competence for the counterfactual and the abstract levels. This seems to show a separation between the familiar content and the other two levels. However, given the very high number of possible combinations for these nine competencies, the other vectors were numerous, showed very small probabilities, and many were equiprobable. We thus decided to use vector classes based on the types of competencies identified in our model. The a posteriori matrix based on our classification is shown in Figure 2. The left column represents the six vector classes we created. Each content level (familiar (F), counterfactual(CF), abstract(A)) represents a triplet of skills (inhibit (I), generate (G), and manage (M)) for this level, while each skill (inhibit, generate, manage) refers to the same skill in all three levels (familiar, counterfactual and abstract). In the first row, "successful" means all three skills for the corresponding level (or all three levels for the corresponding skill) are mastered, regardless of the performance in the other two levels or skills.

For example, being successful in the familiar class may refer to vectors such as "111 111 111", "111 011 111", "111 011 011" or even "111 000 000". In the second row, "failed" means the opposite: all three skills for the corresponding level (or all three levels for the corresponding skill) are not mastered, irrespective of a learner's performance in the other two levels or skills. The "only successful" row means that, of all three skills or levels, only the corresponding one is entirely (111) mastered. For example, being only successful in the familiar class may come from the vector "111 011 011" or even "111 000 000", but not "111 111 010" or "111 000 111". The "Only failed" row refers to the opposite situation: only the corresponding skill or level triple is failed (at least one zero in the triplet), while the two other triplets are fully mastered (111).

	Skill Class Chunk Probabilities			
	Successful	Failed	OnlySuccessful	OnlyFailed
Familiar	0,95	0,05	0,55	0,00
CausalCounterFactual	0,30	0,70	0,00	0,10
Abstract	0,37	0,63	0,01	0,03
Inhibit	0,39	0,61	0,07	0,01
Generate	0,40	0,60	0,10	0,03
Manage	0,41	0,59	0,10	0,02

Figure 2: Skill Class Chunk Probabilities

Results for the different difficulty levels point to a graduation in performance. As can be expected, results show that the "Familiar" level stands out in both "Successful" and "Only Successful" categories: this level is seldom failed and is usually the only one mastered by learners. On the other opposite, the 'Causal Counterfactual' level is the level most often failed and exclusively failed ('Only failed') by learners. While the difference in mastery level is not as pronounced as in the 'Familiar' case, it is nonetheless noteworthy, as both establish clear upper and lower boundaries for

learning and performance. For instance, while the probability that learners exclusively fail the 'Familiar' case is close to zero, the same goes for the probability that learners exclusively master the 'CausalCounterFactual' level. As for the in-between 'Abstract' level, results show that it is nonetheless difficult to master, closer to the 'CounterFactual' level. Finally, results for the different skills ('Inhibit', 'Generate', 'Manage') are inconclusive, as no clear skill pattern can be found across the different difficulty levels.

Validation for the BN and Underlying Psychological Model

Goodness of Fit. We opted for the item pairwise χ^2 measure (Chen, de la Torre, and Zhang 2013) as an absolute indicator of goodness of fit for the CDM model. This measure indicates that the model is inadequate if the p-value of the maximal item pairwise χ^2 measure is above the 0.01 significance level (Groß, Robitzsch, and George 2016). The χ^2 test results were χ^2=33.95 with p-value = 6.79×10^{-7}, which is clearly below any significance threshold and thus indicates that the present model is adequate.

Guess, Slip and Item Discrimination Index (IDI). Results for guess, slip, and item discrimination index parameters are shown in Figure 3. First, we noted a high guess and low IDI for both MTT with familiar content. This can be explained by the fact that a biconditional interpretation of the major premise leads to the correct answer to the MTT, regardless of conditional skills. It is also interesting to note that all items involving few alternatives have a lower IDI than their many alternatives counterparts. Beyond these remarks, however, all items have good IDI values. This is especially true for the MPP counterfactual, which suggests that successfully completing this task is the best way to ensure that conditional reasoning has been fully mastered. As discussed below, this finding is consistent with results obtained for marginal skill probabilities.

	Guess		Slip		Item Easin.	Item Discrim.
	Est.	SE	Est.	SE	(G+(1-S))/2	1-G-S
MPP_FFD	0,27	0,04	0,02	0,01	0,06	0,72
MPP_FMd	0,19	0,03	0,05	0,02	0,05	0,76
MPP_CCF	0,00	0,00	0,05	0,02	0,04	0,95
MPP_A	0,17	0,02	0,00	0,00	0,02	0,83
MTT_FFD	0,51	0,06	0,09	0,02	0,15	0,40
MTT_FMD	0,45	0,05	0,14	0,03	0,14	0,41
MTT_CCF	0,07	0,01	0,19	0,03	0,14	0,74
MTT_A	0,22	0,03	0,09	0,02	0,15	0,69
ACManyAlt	0,08	0,02	0,03	0,01	0,03	0,89
AC_FFA	0,02	0,01	0,25	0,05	0,10	0,73
AC_CCF	0,02	0,01	0,07	0,02	0,05	0,91
AC_A	0,01	0,00	0,11	0,03	0,07	0,87
DA_FMA	0,10	0,02	0,04	0,02	0,05	0,86
DA_FFA	0,03	0,01	0,20	0,05	0,03	0,77
DA_CCF	0,11	0,02	0,04	0,02	0,15	0,85
DA_A	0,05	0,02	0,04	0,02	0,12	0,91
α	0,14	0,02	0,09	0,02	0,08	0,77

Figure 3: Item parameters estimation

Tetrachoric Correlations. Tetrachoric correlations between skills are shown in Figure 4. Based on our sample size, correlation scores over 0.33 are considered significant with α = 0.05 (Guilford and Lyons 1942). Using this criterion for this analysis, one pattern that stands out is that skills with familiar content correlate highly with other skills of the same content level, but not with counterfactual and abstract level skills. However, counterfactual level skills correlate well with themselves and abstract level skills, and vice versa. Overall, these findings suggest a clear separation between the familiar level of content and the other two levels.

	I-F	G-F	M-F	I-CF	G-CF	M-CF	I-A	G-A	M-A	α
I-F	1	0.85	0.9	-0.06	0.05	0.04	0.19	0.15	0.15	0.36
G-F	0.85	1	0.78	0.05	0.06	0.16	0.2	0.16	0.17	0.38
M-F	0.9	0.78	1	0.22	0.06	0.04	0.2	0.15	0.16	0.39
I-CF	-0.06	0.05	0.22	1	0.64	0.6	0.67	0.5	0.52	0.46
G-CF	0.05	0.06	0.06	0.64	1	0.15	0.31	0.4	0.35	0.34
M-CF	0.04	0.16	0.04	0.6	0.15	1	0.35	0.41	0.43	0.35
I-A	0.19	0.2	0.2	0.67	0.31	0.35	1	0.3	0.31	0.39
G-A	0.15	0.16	0.15	0.5	0.4	0.41	0.3	1	0.34	0.38
M-A	0.15	0.17	0.16	0.52	0.35	0.43	0.31	0.34	1	0.38
α	0.36	0.38	0.39	0.46	0.34	0.35	0.39	0.38	0.38	**0.38**

Figure 4: Skill correlations

Marginal Skill Probabilities. The mean mastery percentage for each skill is shown in Figure 5. The hardest skill to master seems to be "Inhibit counterfactual (I-CF)" (44.3%), which requires learners to inhibit disabling conditions to counterfactual conditional statements. This observation is consistent with the fact that the MPP counterfactual has the highest item discrimination index. However, these findings seem at odds with our initial psychological model. Indeed, the latter considers abstract skills to be the hardest and least mastered ones; consequently, abstract skills should be better indicators of mastery for conditional reasoning than counterfactual skills, not the other way around. With hindsight, one could argue that inhibiting exceptions to a rule known to be false, such as "If I throw ketchup on a shirt, then the shirt will be clean," might prove harder than simply inhibiting an imaginary one. This might also be why performances with counterfactual content are similar to or sometimes worse than performances with abstract content. Finally, here as in the previous analyses, the much higher probability of familiar level skills clearly separates them from the other two levels' skills, as the difference between abstract level skills probabilities and their lower-scoring counterfactual counterpart is not nearly as pronounced.

	I-F	G-F	M-F	I-CF	G-CF	M-CF	I-A	G-A	M-A	α
Estimate	0.977	0.961	0.97	0.443	0.55	0.54	0.68	0.64	0.64	0.71

Figure 5: Skill distribution

Summary of the Results. In summary, the analysis of CDM model data provided two significant findings regarding the underlying developmental theory of our Bayesian Network. First, there is a clear separation between the familiar level of content and the other two levels. Moreover, the hardest skill is not an abstract one, like our psycholog-

ical model assumes, but a counterfactual one, i.e. the "inhibit counterfactual" skill. This surprising result, evidenced both by item discrimination indices and marginal skill probabilities, can be explained by the difficulty of inhibiting exceptions, which are in fact, realistic situations, to a false rule. Consequently, counterfactual-level skills may very well prove harder to master than the Abstract level counterparts.

The Bayesian Knowledge Tracing

Now that the Bayesian Network has been created and validated, Logic-Muse can support the Bayesian Knowledge Tracing process (BKT) (Corbett and Anderson 1994). BKT uses the BN to capture students' knowledge which allows inferring the probability of mastering a skill from a specific response pattern (Conati, Gertner, and Vanlehn 2002). Student performance is the observed variable (item variable nodes), which corresponds to the answer to each item attached to the item-type nodes (as in Figure 1). A random variable Q with a Bernoulli distribution represents these item nodes. Q = 1, meaning that the student answered the item correctly, and $Q = 0$, that his/her answer is incorrect.

The upper level of the bayesian network contains latent skill variable nodes that are not directly observed. Some of them, at the deepest level in the network, are related to item variable nodes. Thus, when the learner responds to an item, the probabilities of mastering the different related skills are updated by Bayesian inference. Hence, for the choice of the next exercise to be presented to the learner, the tutor questions the network to extract a list of skill nodes still poorly mastered. From that list, the tutor selects the following relevant items to be proposed to the learner.

In fact, the BKT is a particular case of Hidden Markov Model where student knowledge is represented as a set of binary variables (the skill is either mastered by the student or not). Observations are also binary: a student gets a problem either right or wrong (Yudelson, Koedinger, and Gordon 2013). However, there is a certain probability (G, the Guess parameter) that the student without the necessary skills will give a correct response. Correspondingly, a student who masters the necessary skills related to an item will generally give a correct response. However, there is a certain probability (S, the Slip parameter) that the student will give an incorrect response. The standard BKT model is thus defined by four parameters (d Baker, Corbett, and Aleven 2008): initial knowledge, learning rate (learning parameters), slip and guess. In general, the BKT applies prior knowledge (L0) and the probability of learning the applied concept (p(T)) to measure the progress of student learning. The process works well in Logic-Muse Intelligent Tutoring System. It has also been successfully used in various systems, including computer programming (Kasurinen and Nikula 2009), reading skills (Beck and Chang 2007), etc.

Combining the BN with a Deep Learning Model to Support Deep Knowledge Tracing

Using a BN sometimes implies manually defining apriori probabilities and manually labeling student interactions with relevant concepts. Also, the binary response data used to model knowledge, observations and transitions impose a limit on the kinds of exercises that can be modeled. The Deep Knowldge Tracing (DKT) has been proposed as an excellent alternative to overcome BKT limits.

Deep Knowledge Tracing

Deep learning is an approach that has been successfully applied in many domains, including images recognition (He et al. 2016), Natural Language Processing (Collobert and Weston 2008) and more recently in education for modeling student knowledge. Among the different deep learning architectures that exist, the one that fits well for knowledge tracing is the recurrent neural network, especially the LSTM (Long Short Term Memory), as it is able to capture the sequential aspect of the data, which is helpful for prediction. In that sense, Deep Knowledge Tracing (DKT) (Piech et al. 2015) uses an LSTM to predict student performance based on the pattern of their sequential responses. DKT observes knowledge at both the skill level and the problem level, observing the correctness of each problem. At any time step, the input layer of the DKT is the student's performance on a single problem of the skill that the student is currently working on. In other words, the skill and correctness of each item are used to predict the correctness of the next item, given that problem necessary skills (Piech et al. 2015). Rather than constructing a separate model for each skill as BKT does, DKT models all skills jointly (Khajah, Lindsey, and Mozer 2016; Tato, Nkambou, and Dufresne 2019). It has been shown that DKT can robustly predict whether or not a student will solve a particular problem correctly given the accuracy of historic solutions (Zhang et al. 2017).

Considering Expert Knowledge While Training the DKT Model

Like other machine learning techniques, the DKT is biased towards the data seen during the training phase (it is a data-driven approach). Therefore, the generalization performance of the model depends on the training data. In the Education domain, a priori expert knowledge can be available but is not always sufficiently accurate (e.g., The initial Bayesian Network in Logic-Muse was built by the domain experts (logicians, reasoning psychologists and computer scientists). Nevertheless, even inaccurate models can provide helpful information that should be considered (Zappone et al. 2018). In general, employing a fully data-driven approach to train deep neural networks requires the acquisition of a massive amount of data, which might not always be practical or realistic due to economic reasons or the complexity of the process it entails. Thus combining a priori expert knowledge and data-driven methods using the attentional mechanism constitutes a suitable approach to the design of hybrid deep learning architecture.

In Logic-Muse Tutoring System, a priori knowledge is incorporated using attention mechanism in a deep learning architecture (Tato, Nkambou, and Dufresne 2019; Tato and Nkambou 2022). The proposed architecture was improved to better manage skills that are difficult or easy to master. This is because the initial DKT fails to generalized well on those marginal skills.

F1score	AC_FMA		DA_FFA		AC_CCF		DA_CCF		AC_A		DA_A		Accuracy
DKT	0.74	0.0	0.79	0.0	0.79	0.0	0.73	0.0	0.80	0.0	0.79	0.0	**0.74**
DKTm	0.70	0.36	**0.80**	**0.47**	0.83	0.54	0.76	0.51	0.85	0.48	0.86	0.46	**0.80**
DKTm+BN	**0.70**	**0.74**	**0.80**	**0.47**	**0.87**	**0.80**	**0.82**	**0.72**	**0.83**	**0.58**	**0.88**	**0.80**	**0.82**

Table 4: The DKT, the DKTm, and the DKKm+BN on selected skills

Skills	N	Average	Standard Dev
MPP_FFD	294	**0,9456**	0,16078
MPP_FMD	294	0,898	0,23726
MPP_CCF	294	**0,907**	0,2394
MPP_A	294	**0,9615**	0,16066
MTT_FFD	294	0,8435	0,26646
MTT_FMD	294	0,7925	0,29985
MTT_CCF	294	0,7494	0,33326
MTT_A	294	0,8401	0,28974
AC_FMA	294	0,424	0,38072
AC_FFA	294	**0,3039**	0,3652
AC_CCF	294	**0,3345**	0,40801
AC_A	294	**0,2823**	0,41038
DA_FMA	294	0,407	0,37389
DA_FFA	294	**0,3027**	0,35081
DA_CCF	294	**0,381**	0,40662
DA_A	294	**0,305**	0,42077

Table 5: Distribution of responses over skills: ceiling (Average < 0.4) and floor (Average > 0.9) skills are in bold.

The Dataset We used the same data collected from the 294 participants who participated in the previous study (Bayesian Model). In our dataset, each line of data represents each participant with its sequential performance on the 48 reasoning problems. The amount of data is very few to train a deep learning model. However, combined with expert knowledge, we will see a substantial difference in the results. The exercises were encoded using directly observable skills, meaning that the questions related to the same skill are encoded with the same Id ($1 \sim 16$). The skills with few data are determined by comparing the average of correct answers obtained for each skill. In Table 5, we averaged all the answers on each skill. Since the LSTM only accepts a fixed length of vectors as the input, we used one-hot encoding to convert student performance into a fixed length of vectors whose all elements are 0 except for a single 1. The single 1 in the vector indicates two things: which skill is involved and if the skill was mastered correctly.

The Results To assess our proposed solutions, we ran three models: the DKT, the DKT where we applied a mask to the loss function (DKTm), and the DKTm with apriori knowledge (DKTm+BN). We used 20% of data for testing and 15% for validation. The BN alone gave 65% of global accuracy. The result is evaluated using the $F1$-score on each skill (treated as two classes - correct and incorrect answers) being predicted and the overall accuracy. The models were evaluated in 20 experiments, and the final results were averaged. Our implementation of the DKTm+BN model in Tensorflow using Keras backend was inspired by the implementation[1] done by Khajah et al. (Khajah, Lindsey, and Mozer 2016). Our code is also available on GitHub[2] for further research.

The results are shown in Table 4. We repeated the experiments 20 times. The last column is the overall accuracy of the model. For each skill, the first column corresponds to the value of the $f1$-score for predicting that the skill was incorrectly answered and the second column to the value of the $f1$-score for predicting that the skill was correctly answered. The best ratio for each skill is in bold. Each time the student answers an exercise, we can predict whether or not he will answer an exercise of each type correctly on his next interaction. As expected, the DKTm (original DKT + generalization on marginal skills) enhanced with BN outperforms all other models on predicting skills with little data.

Using the attention mechanism to incorporate expert knowledge into NN is novel. It can be used in other domains, such as text classification or medicine, where expert knowledge is available. For example, we could think of a classifier using a neural network combined with a rule-based system playing the role of expert knowledge.

Conclusion

In this paper, we presented the student model in Logic-Muse Intelligent Tutoring System, which helps learners develop logical reasoning skills. A bayesian student model was first presented, including its validation using a cognitive diagnosis model trained on data collected from 294 learners (each student solved 48 reasoning problems). The resulting CDM model was also used for initializing the bayesian network for a new learner who goes through a pretest (composed of those 48 items) during its first connection to the system. The bayesian network is the primary support of the bayesian knowledge tracing (BKT) in Logic-Muse. We then explored another approach based on Deep Knowledge Tracing (DKT). Using the same dataset, we trained a particular LSTM capable of considering domain expert prior knowledge represented in the bayesian network. The result is a deep neural network model able to predict the student performance on the next item accurately. We show how the new model (which integrates prior knowledge using an attentional mechanism) outperforms classic DKT. More investigation will be conducted on large datasets. In addition, for the CDM method, we will investigate more data-driven approaches (Xu and Desmarais C. 2018) that could significantly improve the Q-Matrix. This could result in re-training the model for better accuracy.

[1] https://github.com/mmkhajah/dkt

[2] https://github.com/angetato/Deep-Knowledge-Tracing-On-Skills-With-Limited-Data/blob/master/DKT_BN.py

References

Beck, J. E.; and Chang, K.-m. 2007. Identifiability: A fundamental problem of student modeling. In *International Conference on User Modeling*, 137–146. Springer.

Chen, J.; de la Torre, J.; and Zhang, Z. 2013. Relative and absolute fit evaluation in cognitive diagnosis modeling. *Journal of Educational Measurement*, 50(2): 123–140.

Collobert, R.; and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167. ACM.

Conati, C.; Gertner, A.; and Vanlehn, K. 2002. Using Bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction*, 12(4): 371–417.

Corbett, A. T.; and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4): 253–278.

Cummins, D. D.; Lubart, T.; Alksnis, O.; and Rist, R. 1991. Conditional reasoning and causation. *Memory & cognition*, 19(3): 274–282.

d Baker, R. S.; Corbett, A. T.; and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, 406–415. Springer.

De La Torre, J. 2009. A cognitive diagnosis model for cognitively based multiple-choice options. *Applied Psychological Measurement*, 33(3): 163–183.

De Neys, W.; Schaeken, W.; and D'ydewalle, G. 2002. Causal conditional reasoning and semantic memory retrieval: A test of the semantic memory framework. *Memory & cognition*, 30(6): 908–920.

De Neys, W.; Schaeken, W.; and D'Ydewalle, G. 2003. Inference suppression and semantic memory retrieval: Every counterexample counts. *Memory & cognition*, 31(4): 581–595.

George, A. C.; Robitzsch, A.; Kiefer, T.; Groß, J.; and Ünlü, A. 2016. The R package CDM for cognitive diagnosis models. *Journal of Statistical Software*, 74(2): 1–24.

Gilovich, T.; Griffin, D.; and Kahneman, D. 2002. *Heuristics and biases: The psychology of intuitive judgment*. Cambridge university press.

Groß, J.; Robitzsch, A.; and George, A. 2016. Cognitive diagnosis models for baseline testing of educational standards in math. *Journal of Applied Statistics*, 43(1): 229–243.

Guilford, J. P.; and Lyons, T. C. 1942. On determining the reliability and significance of a tetrachoric coefficient of correlation. *Psychometrika*, 7(4): 243–249.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kasurinen, J.; and Nikula, U. 2009. Estimating programming knowledge with Bayesian knowledge tracing. In *ACM SIGCSE Bulletin*, volume 41, 313–317. ACM.

Khajah, M.; Lindsey, R. V.; and Mozer, M. C. 2016. How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416*.

Markovits, H. 2013. The development of abstract conditional reasoning. In *The development of thinking and reasoning*, 83–104. Psychology Press.

Markovits, H.; and Vachon, R. 1989. Reasoning with contrary-to-fact propositions. *Journal of Experimental Child Psychology*, 47(3): 398–412.

Nkambou, R.; Mizoguchi, R.; and Bourdeau, J. 2010. *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, 505–513.

Ravand, H.; and Robitzsch, A. 2015. Cognitive diagnostic modeling using R. *Practical Assessment, Research, and Evaluation*, 20(1): 11.

Tato, A.; and Nkambou, R. 2022. Infusing Expert Knowledge Into a Deep Neural Network Using Attention Mechanism for Personalized Learning Environments. *Frontiers in Artificial Intelligence*, 5(921476): 1–19.

Tato, A.; Nkambou, R.; and Dufresne, A. 2019. Hybrid Deep Neural Networks to Predict Socio-Moral Reasoning skills. In *Proceedings of the 12th International Conference on Educational Data Mining*, 623–626.

Thompson, V. A. 1994. Interpretational factors in conditional reasoning. *Memory & cognition*, 22(6): 742–758.

Xu, P.; and Desmarais C., M. 2018. An Empirical Research on Identifiability and Q-matrix Design for DINA model. In *Proceedings of the 11th International Conference on Educational Data Mining*, 438–443. IEDMS.

Yudelson, M. V.; Koedinger, K. R.; and Gordon, G. J. 2013. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, 171–180. Springer.

Zappone, A.; Di Renzo, M.; Debbah, M.; Lam, T. T.; and Qian, X. 2018. Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks towards wireless systems optimization. *arXiv preprint arXiv:1808.01672*.

Zhang, L.; Xiong, X.; Zhao, S.; Botelho, A.; and Heffernan, N. T. 2017. Incorporating rich features into deep knowledge tracing. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, 169–172. ACM.