KBAI Assignment-3

[Arti Chauhan: Nov-5 2016]

Problem background

This paper is motivated by my personal experience of helping my 9 year old prepare for SSAT (Entrance Exam for independent schools) last year. As any dedicated parent, my husband and I chalked out a study plan for her, setting aside few hours every weekend to help her with concepts she was struggling with and gathering all possible online resources, which included buying a tutoring program for standardized tests. While this program was very helpful in providing us insight into her time management skills, proficiency level on different concepts, practice test score, it lacked something very fundamental as a tutor – a human-teacher's appeal.

- For initial few weeks she was quite motivated to use the tutoring program on almost daily basis but soon became disengaged. Tutor would provide feedback (mostly canned) and words of encouragement, but it became too monotonous for her due to lack of interaction.
- Moreover, I found several instances where her test-score was perfect but there were 'holes' in her knowledge. Tutoring program had limited capability to track and fill those gaps.

In following section I will discuss design of a human-like Tutor that would help upper-elementary student prepare for standardize test. Agent's design will address above mentioned deficiencies and leverage **Explanation Based Learning (EBL**) to dynamically update its knowledge. It will use NLP (which also uses EBL at its core) to allow conversation between student and Agent and learn student's state of mind.

Key requirements/challenges for Tutor Agent

1. **Deep learning**: A good teacher has an innate capability to develop love of learning in children. They may not be as efficient as computers in grading and tracking each student' score, but they know their students inside out and hence are able to connect with them at social and emotional level. This develops high level of student engagement and deep learning. How can we develop this trait in our Agent?

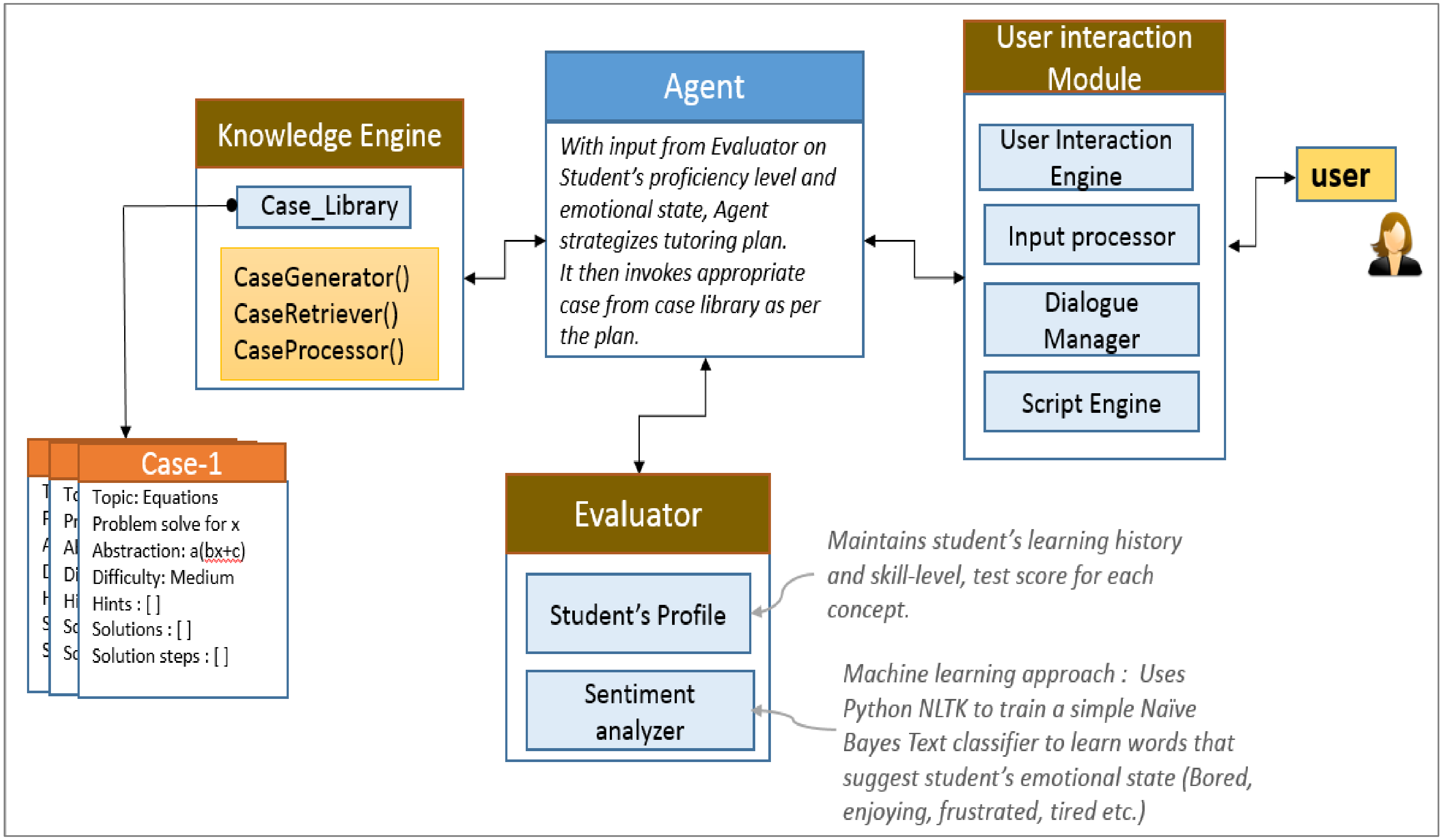
2. **Knowledge abstraction and Practice**: As in any classroom environment, learning occurs through information acquisition, reusing and adapting that knowledge for assigned activities. Agent should be able to not only teach a concept to a student but also give student ample opportunity to practice adaptation of that knowledge for similar or related problems.

3. **Timely and right amount of feedback:** In addition to student's cognition, his /her emotional state (especially for young student) plays a key role. Human teacher allows student to attempt the problem and jumps in at the right time with just the right amount of amount of information (hints) to avoid student feeling frustrated , yet not revealing the answer. Jumping in too soon or too late is detrimental to effectiveness of learning.
   How can Agent guess student's skill level and emotional state and adjust its tutoring strategies accordingly?

4. **Time Management**: For any standardized test, time management is key. How can Agent teach this skill to young students?

Design and Implementation

Figure below provides a high level view of Tutoring Agent. It has 3 main modules – Knowledge Engine, Evaluator, User-Interaction module


Knowledge Engine
Case_Library
CaseGenerator()
CaseRetriever()
CaseProcessor()
Agent
With input from Evaluator on Student's proficiency level and emotional state, Agent strategizes tutoring plan.
It then invokes appropriate case from case library as per the plan.
User interaction Module
User Interaction Engine
Input processor
Dialogue Manager
Script Engine
user
Case-1
Topic: Equations
Problem solve for x
Abstraction: a(bx+c)
Difficulty: Medium
Hints : [ ]
Solutions : [ ]
Solution steps : [ ]
Evaluator
Student's Profile
Sentiment analyzer
Maintains student's learning history and skill-level, test score for each concept.
Machine learning approach : Uses Python NLTK to train a simple Naïve Bayes Text classifier to learn words that suggest student's emotional state (Bored, enjoying, frustrated, tired etc.)

1. **Knowledge Engine:**

   This module relies on Case Based Reasoning and Explanation based learning. A CBR system reasons by first retrieving a relevant prior case from its memory of cases, reusing or revising (adapting) the solution of the old case to solve the new problem and retaining the updated solution. Agent uses EBL to connect previously known concepts, which enables it to transfer knowledge from a prior situation to a new situation. (Goel & Joyner, 2016, p. 223)

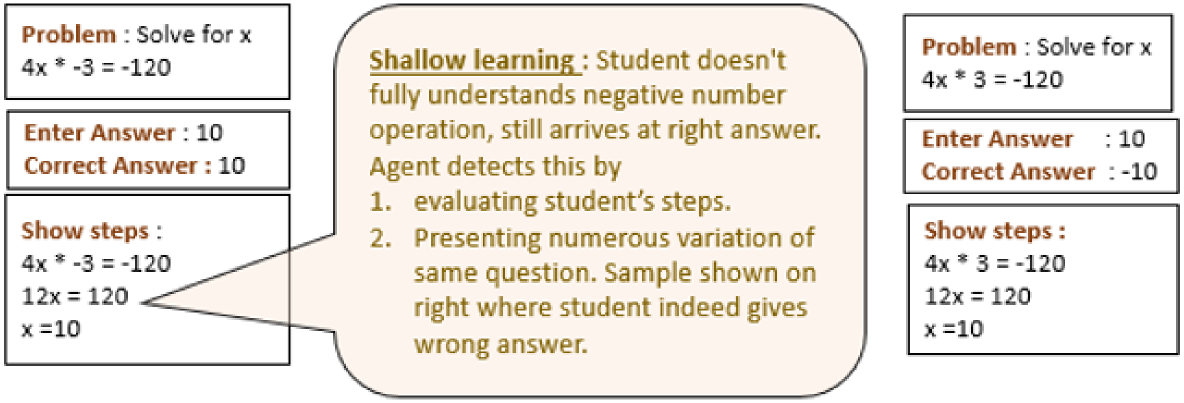
1. *Agent's knowledge Engine has a case-library which is organized by domain/topics.*
2. *Cases stored in this library entails information about various steps required to solve a problem, different ways to solve the given problem, hints for the student, concept abstraction and difficulty level.*
3. *Agent uses Explanation based learning to generate and manage cases in following manner*

a) ***Generating cases from a pre fed sample case:***

Agent is prebuilt with basic elementary math rules and few sample problems on topics shown in Figure-2 below. Agent abstracts the problem by replacing numeric values with variables and generates new cases using EBL which enables it to connect previously known concepts. This approach not only reduces the human–cost involved in pre populating Agent's case-library but also reduces the number of cases generated (and hence less cases stored and retrieved), thus improving the efficiency of the Agent. Figure-2 and 3 below illustrates this concept.

Input case: Solve for x: 4(2x+3) =20
Step 1: Agent performs abstraction of this case (replaces constants with variable a, b, c, d) => a (bx+c) =d
Step 2: Generate variations of this problems by utilizing prior knowledge about commutative property and negative number operations concepts.

Creating numerous variation of this question by variablizing the operands and changing structure of the equation, ensures that student has deep understanding of the concept and avoids shallow learning as seen in case below.



Problem : Solve for x
4x * -3 = -120

Enter Answer : 10
Correct Answer : 10

Show steps :
4x * -3 = -120
12x = 120
x =10

Shallow learning : Student doesn't fully understands negative number operation, still arrives at right answer. Agent detects this by
1. evaluating student's steps.
2. Presenting numerous variation of same question. Sample shown on right where student indeed gives wrong answer.

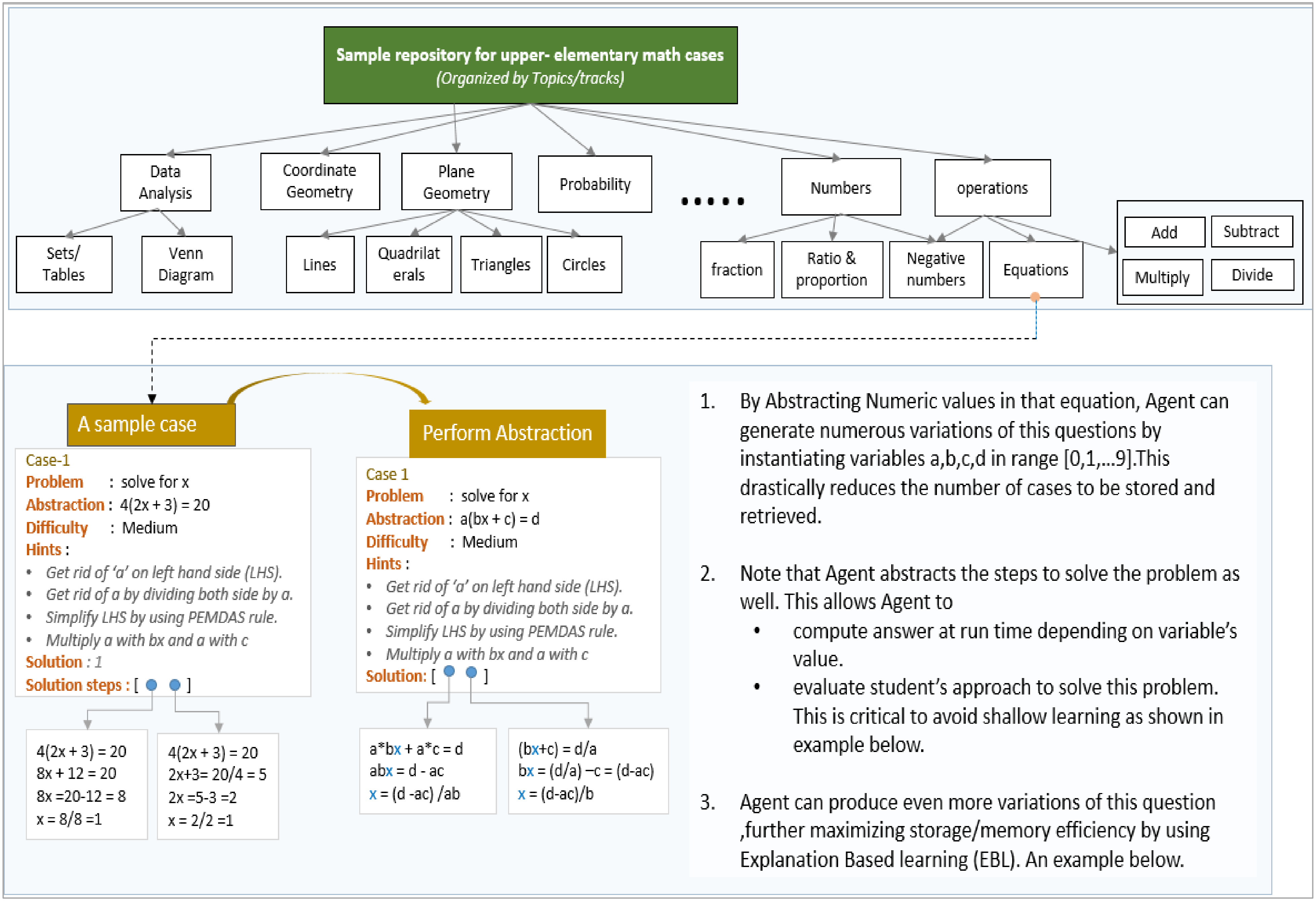
Problem : Solve for x
4x * 3 = -120

Enter Answer : 10
Correct Answer : -10

Show steps :
4x * 3 = -120
12x = 120
x =10

b) ***Generating cases from a novel problem:***

Figure-4 below illustrates this scenario. Agent doesn't have a base case here. It sees a triangle and a circle in new problem so it probes its prior knowledge to extract triangle and circle concepts and builds a graph using EBL to transfer the knowledge to reason over the problem.



Case-1
Problem : solve for x
Abstraction : 4(2x + 3) = 20
Difficulty : Medium
Hints :

- *Get rid of 'a' on left hand side (LHS).*
- *Get rid of a by dividing both side by a.*
- *Simplify LHS by using PEMDAS rule.*
- *Multiply a with bx and a with c*

Solution : *1*
Solution steps : [● ●]

4(2x + 3) = 20
8x + 12 = 20
8x =20-12 = 8
x = 8/8 =1

4(2x + 3) = 20
2x+3= 20/4 = 5
2x =5-3 =2
x = 2/2 =1

Case 1
Problem : solve for x
Abstraction : a(bx + c) = d
Difficulty : Medium
Hints :

- *Get rid of 'a' on left hand side (LHS).*
- *Get rid of a by dividing both side by a.*
- *Simplify LHS by using PEMDAS rule.*
- *Multiply a with bx and a with c*

Solution: [● ●]

a*bx + a*c = d
abx = d - ac
x = (d -ac) /ab

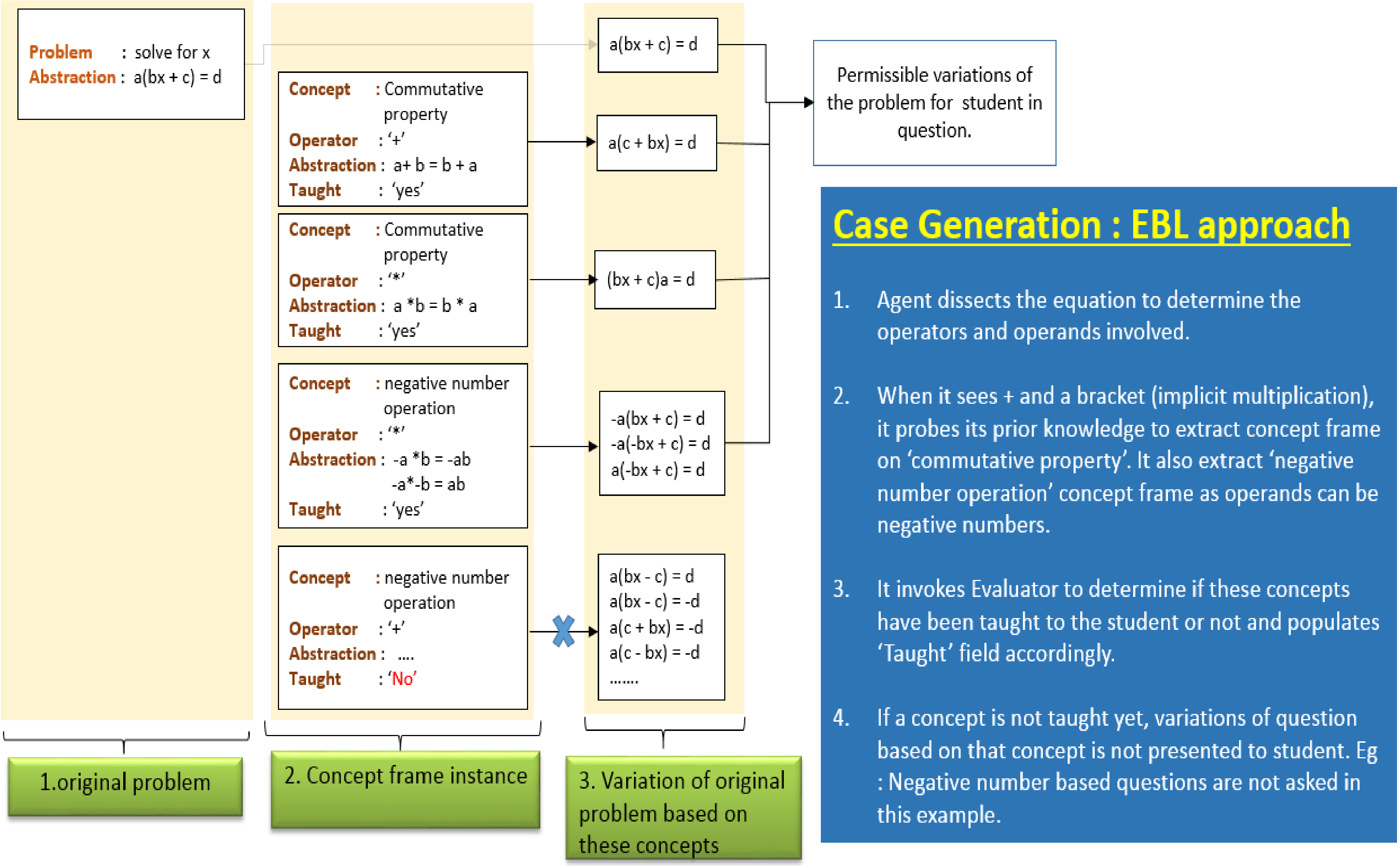
(bx+c) = d/a
bx = (d/a) –c = (d-ac)
x = (d-ac)/b

1. By Abstracting Numeric values in that equation, Agent can generate numerous variations of this questions by instantiating variables a,b,c,d in range [0,1,...9].This drastically reduces the number of cases to be stored and retrieved.

2. Note that Agent abstracts the steps to solve the problem as well. This allows Agent to
   - compute answer at run time depending on variable's value.
   - evaluate student's approach to solve this problem. This is critical to avoid shallow learning as shown in example below.

3. Agent can produce even more variations of this question ,further maximizing storage/memory efficiency by using Explanation Based learning (EBL). An example below.

Figure-2: Problem abstraction


Problem : solve for x
Abstraction : a(bx + c) = d
Concept : Commutative property
Operator : '+'
Abstraction : a+ b = b + a
Taught : 'yes'
Concept : Commutative property
Operator : '*'
Abstraction : a *b = b * a
Taught : 'yes'
Concept : negative number operation
Operator : '*'
Abstraction : -a *b = -ab
-a*-b = ab
Taught : 'yes'
Concept : negative number operation
Operator : '+'
Abstraction : ....
Taught : 'No'
a(bx + c) = d
a(c + bx) = d
(bx + c)a = d
-a(bx + c) = d
-a(-bx + c) = d
a(-bx + c) = d
a(bx - c) = d
a(bx - c) = -d
a(c + bx) = -d
a(c - bx) = -d
....
Permissible variations of the problem for student in question.
1.original problem
2. Concept frame instance
3. Variation of original problem based on these concepts

Case Generation : EBL approach

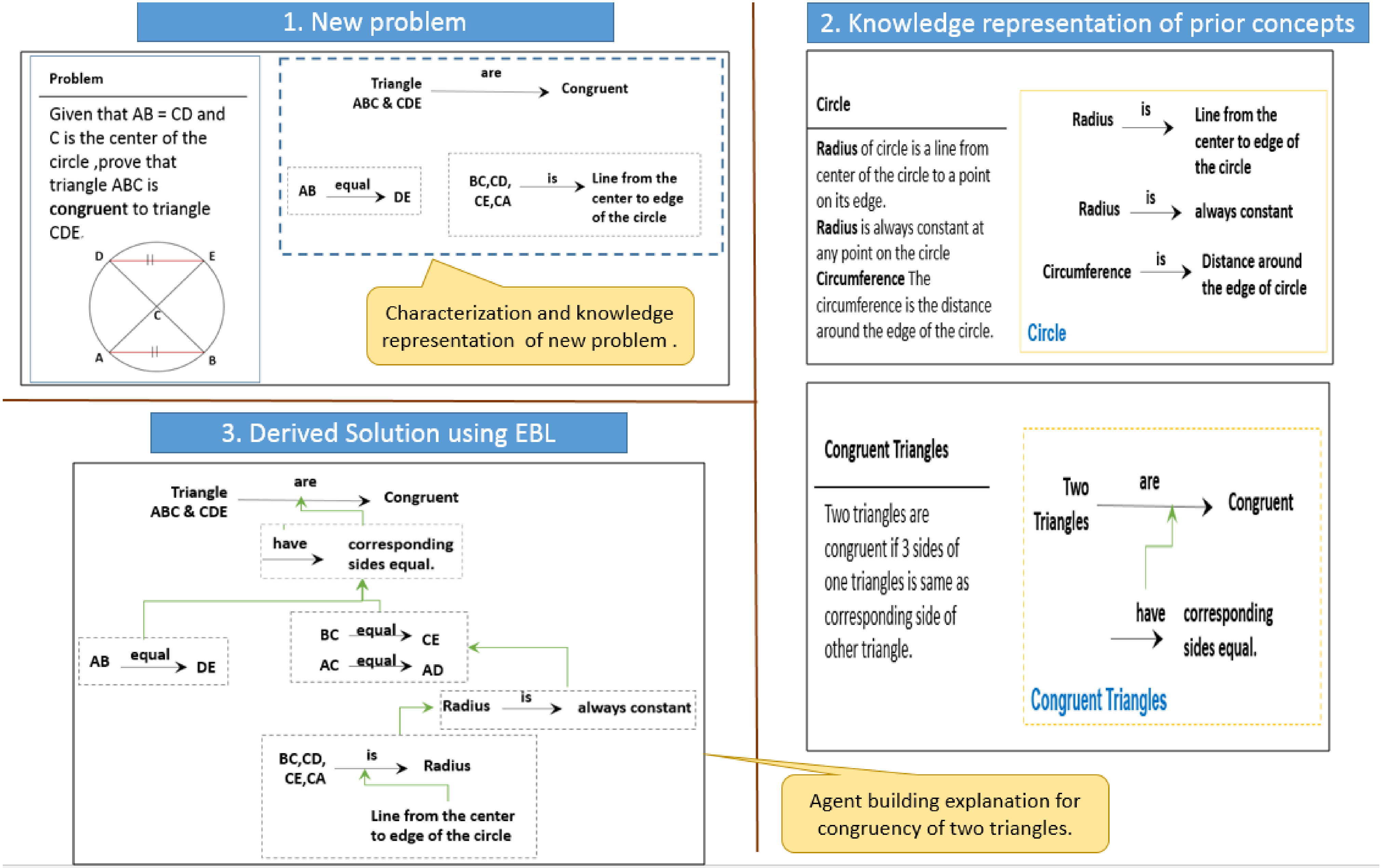
1. Agent dissects the equation to determine the operators and operands involved.

2. When it sees + and a bracket (implicit multiplication), it probes its prior knowledge to extract concept frame on 'commutative property'. It also extract 'negative number operation' concept frame as operands can be negative numbers.

3. It invokes Evaluator to determine if these concepts have been taught to the student or not and populates 'Taught' field accordingly.

4. If a concept is not taught yet, variations of question based on that concept is not presented to student. Eg : Negative number based questions are not asked in this example.

Figure-3: Case generation from a base case using EBL



1. New problem

Problem

Given that AB = CD and C is the center of the circle ,prove that triangle ABC is **congruent** to triangle CDE.

Characterization and knowledge representation of new problem .

2. Knowledge representation of prior concepts

Circle

Radius of circle is a line from center of the circle to a point on its edge.

Radius is always constant at any point on the circle

Circumference The circumference is the distance around the edge of the circle.

Congruent Triangles

Two triangles are congruent if 3 sides of one triangles is same as corresponding side of other triangle.

3. Derived Solution using EBL

Agent building explanation for congruency of two triangles.

Figure-4: Case generation from a novel problem using EBL

(Format adapted from Goel & Joyner, 2016, p. 225)

2. **Evaluator :**

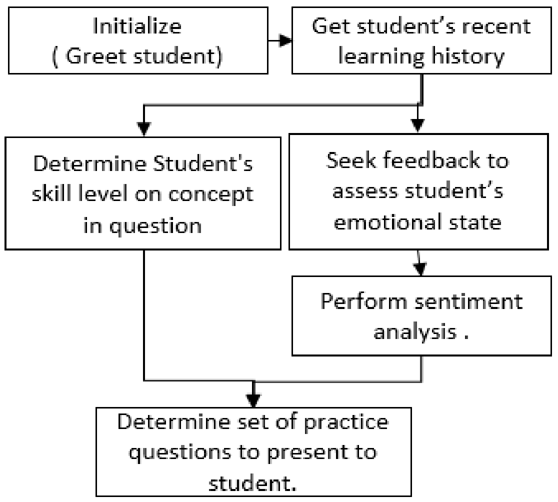
This module is tasked with evaluating student's skill level and emotional state and strategizing tutoring plan accordingly.

- It maintains each student's profile, history of concepts he has learned and corresponding proficiency in that area. It provides a 'progress-card' view for each student which shows number of tests taken, test-score, test-time on each concept, student's feedback on questions he missed etc. and suggested path for improvement.
- While one's emotional state can be judged by one's tone, body language and facial expression, in this design I propose a simple approach based on text feedback from student. This approach leverages machine learning to perform a sentiment analysis on student's feedback. Evaluator works with User-interaction module to seek input from student to determine his emotional state. A sample case is presented below that describes how Agent might accomplish this task.

Sample case : Agent trying to assess student's emotional state and taking appropriate action.

Agent : Hello Tanya , how are you today ?
Student : I am good.
Agent : How do you feel about equations we learned yesterday ?
Student : I am confused.
Agent : What is it that you found confusing ?
Student : I am not clear about negative numbers when solving equation.
Agent : No worries. Lets go over negative numbers one more time Are you ready ?
Student : yes

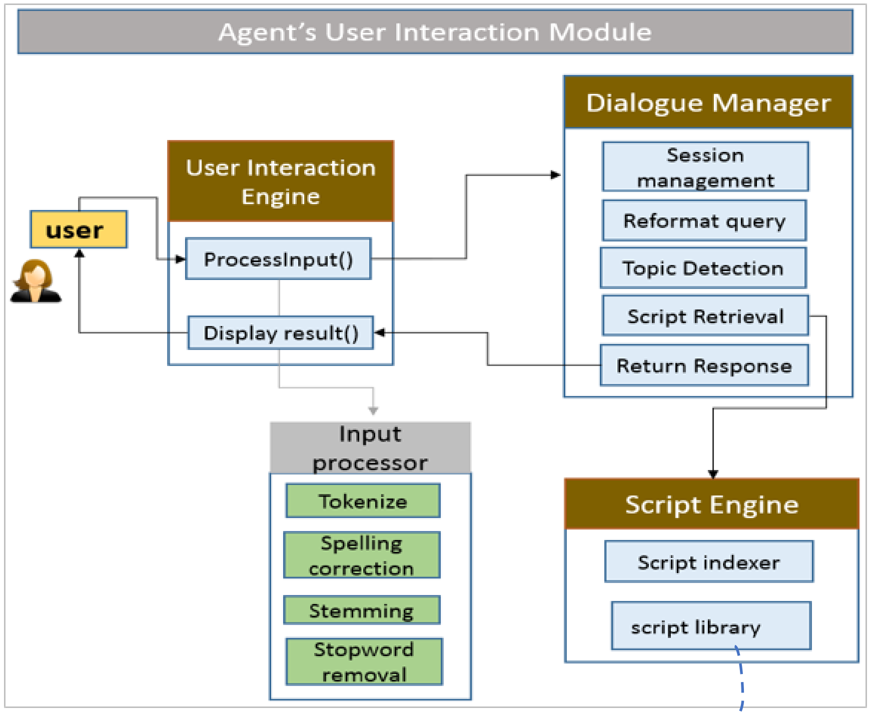
Sample script for dialogue described on left


Initialize
(Greet student)
Get student's recent learning history
Determine Student's skill level on concept in question
Seek feedback to assess student's emotional state
Perform sentiment analysis .
Determine set of practice questions to present to student.

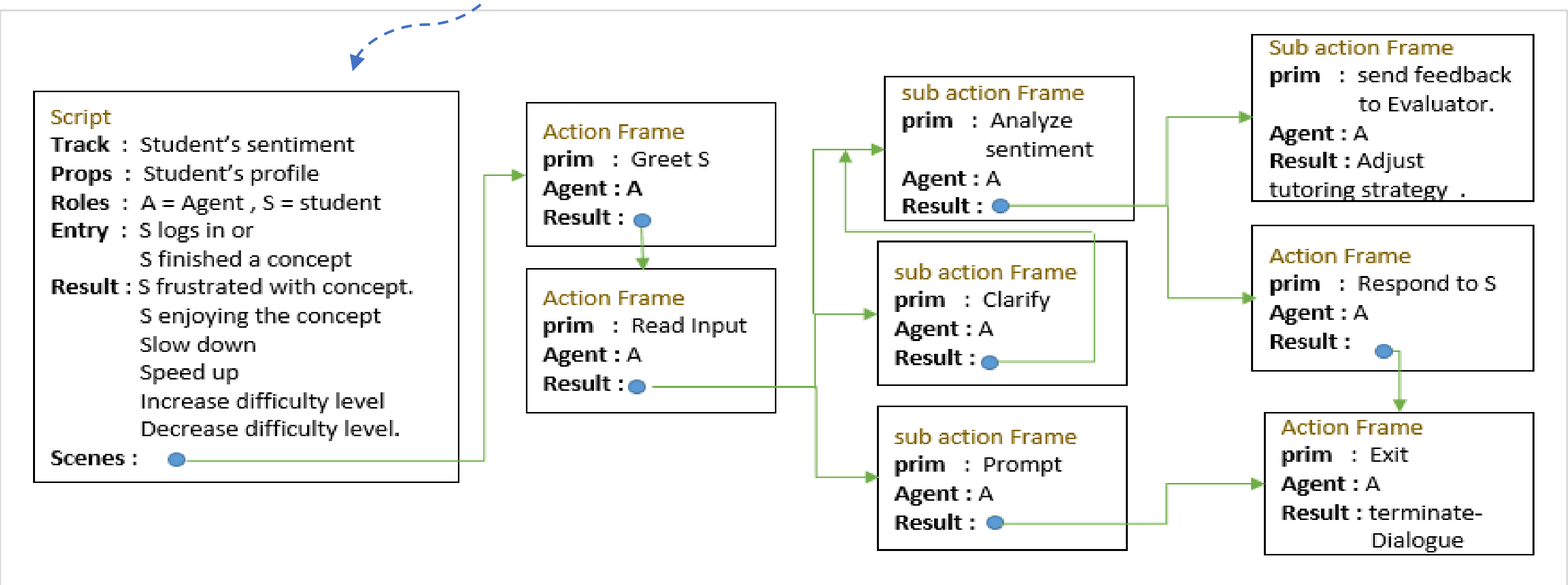
Based on above information, Agent adjusts its tutoring plan. It may decrease the level of difficulty on questions asked if student is student is struggling and may suggest to review the concept or it may ask student to explain the concept and evaluate it against its knowledge to determine gap in student's understanding or it may decide to stick with original plan but provide frequent feedback to help student make progress. Building this logic can be accomplished by incorporating production rule or a plan that lets Agent determine what steps to take for a given skill-level and emotional state.

3. **User-Interaction module**

This module is a topic in itself and was addressed in detail in my Mid-term paper. Below I have described its main functionality, adapted from that paper [3]



1. *User Interaction module provides a natural language interface to its user. It calls its 'input-processor' sub-module which performs tokenization, spelling correction, stemming and stop word removal.*
2. *Processed input is then passed to Dialogue-Manager. Dialogue-Manager binds the user to a unique dialogue/ session Id. This session keeps track of requests/responses exchanged with user, which enables Agent to learn context of the conversation, resolve referent and detect topic being discussed.*
3. *Dialogue-Manager invokes Script Engine to retrieve relevant script and returns the response listed in retrieved script to User Interaction module, which then displays it to the user.*
4. *Script Engine maintains a script repository, organized by Track/Topic. It finds the script that shows highest similarity to the input and returns it to Dialogue-Manager.*
5. *If no script is found, user is asked to clarify the question. If after N attempts, Agent is still not able to find a relevant script, a default response (e.g.: an apology message) is displayed to the user and Agent tries to make up for this shortcoming by suggesting a topic or presenting information that user might find helpful. Having picked some cues about the context during the course of conversation, Dialogue-Manager makes educated guesses on what topics to suggest or what information to present.*



Script
Track : Student's sentiment
Props : Student's profile
Roles : A = Agent , S = student
Entry : S logs in or
S finished a concept
Result : S frustrated with concept.
S enjoying the concept
Slow down
Speed up
Increase difficulty level
Decrease difficulty level.
Scenes : ●

Action Frame
prim : Greet S
Agent : A
Result : ●

Action Frame
prim : Read Input
Agent : A
Result : ●

sub action Frame
prim : Analyze sentiment
Agent : A
Result : ●

sub action Frame
prim : Clarify
Agent : A
Result : ●

sub action Frame
prim : Prompt
Agent : A
Result : ●

Sub action Frame
prim : send feedback to Evaluator.
Agent : A
Result : Adjust tutoring strategy .

Action Frame
prim : Respond to S
Agent : A
Result : ●

Action Frame
prim : Exit
Agent : A
Result : terminate-Dialogue

__Clarify__ : if agent is not able to find an appropriate script, it asks student to further clarify his input. .
__Prompt__ : Agents tries to engage user by prompting on topic related to current context of conversation.

Figure-6: A sample script from script-library

How does this Agent addresses challenges listed above

Deep learning: Agent is able strike a conversation with the student and study his emotional state. Like any human-teacher, Agent takes into account student's mastery on a concept (via keeping track of students' tests-score, response time, quality of response etc.) and his emotional state to decide the lesson plan for the day. As shown in example above, Agent creates numerous variation of same question/concept by variablizing the operands and changing structure of the problem statement, which avoids shallow learning.

Knowledge abstraction and Practice: Unless student is given ample opportunity to practice the concept, learning may be short-lived. Studies have shown that when student is asked to explain a concept (rather than just passively listening to an explanation), higher level of cognition is fired. Agent doesn't just relies on final answer. In addition to giving practice tests, Agent asks student to show his work to detect any holes in his knowledge. Student is asked to input steps he took to solve the problem in user-interaction window. Agent has all possible solution approaches for a given problem in its case memory (Fig 2), against which it compares student's response and provides appropriate feedback.

Timely and right amount of feedback: This is a tricky one. In a typical tutoring system student has control of over hint option, which can easily be misused. Student can keep asking for hints until Agent spits out the answer. To discourage this behavior, Agent allocates a fixed 'hint' quota for a set of questions, which encourages student to use it wisely and not exhaust it on one question. In addition, Student is allowed to redeem unused hint quota for some incentive (Eg: extra points). Based on difficulty level of a question (relative of student's skill level), Agent will determine hint-quota and may prompt Student if student is not making progress.

Time Management: Agents keeps a history of student's response time to test questions. It helps Agent to determine which concepts require more practice in order to build student's speed and create a lesson plan accordingly.

Reference

1. Goel, A. & Joyner, D. (2016). KBAI ebook: Knowledge Based Artificial Intelligence. Retrieved from https://files.t-square.gatech.edu/access/content/group/gtc-c01a-0bd6-5b67-9c88176eb86d0533/KBAI%20ebook/kbai_ebook.pdf

2. Cognitive Tutors: Technology Bringing Learning Science to the Classroom (Kenneth R. Koedinger, Albert Corbett)

3. Chauhan, Arti (2016) , CS 7637 Mid-term Examination– Design of a Conversational Agent, unpublished paper

4. Doug Lenat (2015), Computers with common sense , https://www.youtube.com/watch?v=2w_ekB08ohU