Cyberbook

* Finalize content

1. Use watson and interface to teach simsudent -production model. Drain expert model for model tracing is missing. CTAT box -> Selection Action I ‘3x’ entered action -> model tracer which needs production full of knowledge. To make production , one idea is teaching simstudent. Make production is my job. I need Cognitive task analysis and think aloud
2. What kind of data should we collect from cyberbook?
3. Reinforcement learning.
4. WS : One of the Evidence-based improvement, Overall quality efficiency/accuracy/quality goes up;. WS in the context of adaptive courseware. -> !lots of cognitive tutors in courseware. Not one.

* Open Edx documentation : <https://docs.google.com/document/d/1g-d0udj_QHJpJKW4_WlUgvYDbjBH1oeW_QcLH_XfyjE/edit>
* ‘write domain dependent knowledge’ document.<https://docs.google.com/document/d/1XOG2qMyoXE6mnOdbuvj2mLEb3wWiK1s3I4iB6M-eNIo/edit#heading=h.s8yn3tjcefx5>
* Watson documentation <https://docs.google.com/document/d/1b9d_B1qTGYkiZYS-6ntMzo22Z2hUHwEsi3laqF8l2TU/edit#>
* Mocha : <https://mocha.education.tamu.edu:8443/SimStudentServlet/SimStWebTutor/login.html>

12/06

* Goal is to Design operators(연산자) for my interfaces/ Feature predicates (when to use)

Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2015). [Teaching the Teacher: Tutoring SimStudent leads to more Effective Cognitive Tutor Authoring](http://www.simstudent.org/publications/ijaied2014-1). International Journal of Artificial Intelligence in Education, 25, 1-34.

* Model tracing is a heuristic(발견적인, 학습적인) version of plan recognition (Burton and Brown 1982; London and Clancey 1982) that attempts to identify cognitive skills in the expert model that sufficiently reproduce problem-solving steps performed by students (Anderson and Pelletier 1991).
* Knowledge tracing longitudinally models a student’s mastery of individual cognitive skills (Corbett et al. 2000; Koedinger and Corbett 2006).
* authoring a cognitive tutor comes down to authoring two domain dependent components: (1) the graphical user interface (GUI) called a Tutoring Interface for students to show their work, and (2) the expert model representing skills to be learned with hint and error messages associated to each skill.

* SimStudent: A Synthetic Student That Learns an Expert Model
* an expert model is represented as a set of production rules. Each production rule represents an individual cognitive skill required to perform a particular tutoring step in the Tutoring Interface.

12/18

* **Making files to convert CTAT tutor to model tracing tutor.**
* Following ‘write domain dependent knowledge’ document.<https://docs.google.com/document/d/1XOG2qMyoXE6mnOdbuvj2mLEb3wWiK1s3I4iB6M-eNIo/edit#heading=h.s8yn3tjcefx5>
* Create ‘geometrytutor’
* Create ‘spark.lentghTowpoints’ subpackage. (seoyeon is package)
* Rename : right click -> refracter-> rename.
* The tutor that I’m working on (act 39\_3) needs constraint file. Because it doesn’t have to have a feature predicate file (which represents ‘if~’part. Ex. if the box needs number, then do this. But all the boxes were just number. And there was no need to classify any boxes.
* I changed the ID of the boxes in CTAT tutor to a1, b1, c1, d1/ a2, b2, / a3, b3/ a4/ a5.
* Constraint : constraint means the location of the boxes. First, we made ‘sameRow’ constraint file which means that boxes with same number are in the same row. But we changed it to 5 constraint file - isFirstline, isSecondline, isThirdline, isFourthline, isFifthline, which are asking if the box is in the first row or not, so on.
* Constraint file should take two arguments. But I don’t have to write two functions. Just care about 1 arguments, it’s okay.
* Samerow constraint

package spark.lengthTwopoints;

import java.util.Vector;

import edu.cmu.pact.miss.WMEConstraintPredicate;

import edu.cmu.pact.miss.userDef.topological.table.TableConstraint;

import jess.Fact;

import jess.JessException;

import jess.Rete;

import jess.Value;

public class Constraint extends WMEConstraintPredicate{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public Constraint()

{

setArity(2);

setName("sameRow");

}

//try-catch means if there’s an error, put it into catch.

public boolean sameRow(Fact f1,Fact f2) {

String cell1 = "";

String cell2 = "";

try {

cell1 = f1.getSlotValue("name").toString();

cell2 = f2.getSlotValue("name").toString();

return cell1.charAt(1) == cell2.charAt(1);

} catch (JessException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

@Override

public String apply(Vector args, Rete rete) {

if(sameRow((Fact)args.get(0),(Fact)args.get(1)))

return "T";

else

return null;

}

}

* IsFirstline constraint

package spark.lengthTwopoints;

import java.util.Vector;

import edu.cmu.pact.miss.WMEConstraintPredicate;

import edu.cmu.pact.miss.userDef.topological.table.TableConstraint;

import jess.Fact;

import jess.JessException;

import jess.Rete;

import jess.Value;

public class IsFirstline extends WMEConstraintPredicate{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public IsFirstline()

{

setArity(2);

setName("isFirstline");

}

public boolean isFirstline(Fact f1,Fact f2) {

String cell1 = "";

try {

//cell1 ID looks like 'a1', 'b1'....

cell1 = f1.getSlotValue("name").toString();

int row = (int)cell1.charAt(1);

return row == 1;

} catch (JessException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

@Override

public String apply(Vector args, Rete rete) {

if(isFirstline((Fact)args.get(0),(Fact)args.get(1)))

return "T";

else

return null;

}

}

* UserDefSymbols.java

package spark.lengthTwopoints;

import jess.Rete;

import jess.Userpackage;

public class UserDefSymbols implements Userpackage {

/\*\*

\* Creates a new <code>UserDefSymbols</code> instance.

\*

\*/

public UserDefSymbols() {

}

// Implementation of jess.Userpackage

/\*\*

\* Describe <code>add</code> method here.

\*

\* @param rete a <code>Rete</code> value

\*/

public final void add(final Rete rete) {

// System.out.println("UserDefSymbols for Equation Tutor......");

// Feature predicates

//

// rete.addUserfunction( new CanBeSimplified() );

rete.addUserfunction( new AddOperator() );

rete.addUserfunction( new CopyOperator() );

rete.addUserfunction( new RootOperator() );

rete.addUserfunction( new SquareOperator() );

rete.addUserfunction( new SubtractOperator() );

rete.addUserfunction( new IsFirstline());

rete.addUserfunction( new IsSecondline());

rete.addUserfunction( new IsThirdline());

rete.addUserfunction( new IsFourthline());

rete.addUserfunction( new IsFifthline());

}

}

* TypeCheck.java

package spark.lengthTwopoints;

import edu.cmu.pact.miss.FeaturePredicate;

public abstract class TypeCheck extends FeaturePredicate{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public static final int TYPE\_NUMBER=1;

public static Integer valueTypeCheckerForMyDomain( String value ) {

int valueType=0;

if (isNumber(value)) {

valueType = TYPE\_NUMBER;

}

return new Integer(valueType);

}

private static boolean isNumber(String line){

return line.matches("[+-]?\\d\*(\\.\\d+)?");

}

}

* RootOperator.java

package spark.lengthTwopoints;

import java.util.Vector;

public class RootOperator extends TypeCheck{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public RootOperator()

{

setArity(1);

setName("Root");

setReturnValueType(TYPE\_NUMBER);

setArgValueType(new int[] {TYPE\_NUMBER});

}

@Override

public String apply(Vector args) {

// TODO Auto-generated method stub

int number = Integer.parseInt((String)args.get(0));

double result = Math.sqrt(number);

return result+"";// ‘+””’ means that the result would turn into string.

}

public static void main(String[] args) {

RootOperator op = new RootOperator();

Vector arg = new Vector();

arg.add("5");

String result = op.apply(arg);

System.out.println(" Square root : "+result);

}

}

* CopyOperator.java

package spark.lengthTwopoints;

import java.util.Vector;

public class CopyOperator extends TypeCheck {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public CopyOperator()

{

setArity(1);

setName("copy");

setReturnValueType(TYPE\_NUMBER);

setArgValueType(new int[] {TYPE\_NUMBER});

}

@Override

public String apply(Vector args) {

int number = Integer.parseInt((String)args.get(0));

int result = number;

return result+"";

// TODO Auto-generated method stub

}

}

* AddOperator.java

package spark.lengthTwopoints;

import java.util.Vector;

public class AddOperator extends TypeCheck {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

public AddOperator()

{

setArity(2);

setName("add");

setReturnValueType(TYPE\_NUMBER);

setArgValueType(new int[] {TYPE\_NUMBER,TYPE\_NUMBER});

}

@Override

public String apply(Vector args) {

int number = Integer.parseInt((String)args.get(0));

int number2 = Integer.parseInt((String)args.get(1));

int result = number + number2;

return result+"";

// TODO Auto-generated method stub

}

}

* **Teach SimStudent in Mocha server. :** <https://docs.google.com/document/d/1b9d_B1qTGYkiZYS-6ntMzo22Z2hUHwEsi3laqF8l2TU/edit#heading=h.cavkueevejgw>
* Go to CTAT google drive, get html file& .css file, and javafiles that I made.
* Package username should be the same as the server ID.
* My ID is spark, and the password is 1234.
* Before import all the files, do below things.
* In the html file, remove the following line

<script data-silex-static="true" type="text/javascript" src="https://cdn.ctat.cs.cmu.edu/releases/latest/ctatloader.js"></script>

<script data-silex-static="true" type="text/javascript" src="<https://cdn.ctat.cs.cmu.edu/html-editor/js/ctat/ctat.min.js>"></script>

* In the html file , look for the following pattern, where $$$$$ has to be your tutor name

<link rel="stylesheet" href="Assets/$$$$$-styles.css">

Remove ‘Assets/’ from the line

For example, in my html file MetricTutor.html, I found the following the line

<link rel="stylesheet" href="Assets/MetricTutor-styles.css"> now after removing ‘Assets/’ , the line look like this

<link rel="stylesheet" href="MetricTutor-styles.css">

* Add the following line in the html between ‘</div>’ and ‘</body>’

<script data-silex-static="true" type="text/javascript" src="/SimStudentServlet/SimStWebTutor/js/ctatloader.js"></script>

<script data-silex-static="true" type="text/javascript" src="/SimStudentServlet/SimStWebTutor/js/ctat.min.js"></script>

<script data-silex-static="true" type="text/javascript" src="/SimStudentServlet/SimStWebTutor/js/externalScript.js"></script>

* Import constraint files to the feature predicate blank.
* TypeChecker -> New Type Checker

-> name is classname of TypeChecker javafile.valueTypeCheckerForMyDomain

* When you create the start state, you should put number, enter, number, enter….and click the create start state. -> wait for the message ‘read xml’.
* When you teach SimStudent how to solve the problem, always double click the focus of attention(box or boxes) -> show the step(fill the box) -> write down skill name under the screen, enter.
* Sometimes, SimStudent failed to learn a skill, then run that Operator java file and should get the exact number(result), copy and paste the result to the box.
* When you ask SimStudent ‘do the next step’, you should give it feedback yes/no.
* Even though you succeed to teach all the step, SimStudent is highly likely to fail to replicate all those steps. -> this means SimStudent need more information ( more constraint file, chunk, operator, feature indicate…)

12/19

* **Revise and Make a perfect version of ‘Coordinate Geometry’ course for the pilot study.**
* Green line : there is no skill for the text (or problem..etc.)
* Red line : there is no problem with the same skill. (unmatch)
* I can see many red lines in the math course even the text and the problem have the same skill name. Maybe it’s because of the length, equation, space or something else.

1. Line - 1. recognize the algebraic form of a line (7 word, 22 character, 6 space)
2. Line - 2. determine a y-intercept from y=mx+b (5 word, 21 character, 4 space, -, =, +) : Q1 red line.
3. Line - 2. determine a y-intercept from y=mxb ( exclude +) : text red line.
4. Line - 2. determine a y-intercept from ymx+b ( exclude =) : both red line.
5. Line - 2. determine a y-intercept from ymxb ( exclude +,=) : okay. -> exclude +, = from the skill name. But the link message (when student click wrong answer) doesn’t show up.
6. Line - 3,4 no link.
7. Line - 6. & Squares -2No problem.
8. Writing the equation of a line - no link.
9. Systems of linear equations - 3. No link.
10. Area of triangle - no link.
11. Circles - no link.
12. Ellipses - no link.

To do :

1. how to calculate bkt without the empirical data
2. working on the cognitive tutor with Donna’s video.

12/22

딥러닝 시작서:

<https://espressobook.com/books/481/pages/2246/preview>

* What is cold start problem?

In this paper, we discuss hybrid approaches, using collaborative and also content data to address cold-start - that is, giving recommendations to novel users who have no preference on any items, or recommending items that no user of the community has seen yet.

However, such pure collaborative filtering cannot help in a cold-start setting, where existence of users and items without preference is probable.

* Cold start in bkt?

Nedungadi, P., & Remya, M. S. (2014, October). Predicting students' performance on intelligent tutoring system—Personalized clustered BKT (PC-BKT) model. In *Frontiers in Education Conference (FIE), 2014 IEEE* (pp. 1-6). IEEE.

Cold-start problem refers to the issues arising when new students are faced with new tasks involving a skill for which the student has not received training.

Schatten, C., Janning, R., & Schmidt-Thieme, L. (2015, January). Integration and Evaluation of a Matrix Factorization Sequencer in Large Commercial ITS. In *AAAI* (pp. 1380-1386).

Given the large data availability we had no cold–start problem, which is experienced in MF when not enough data on tasks or on students are available. The task cold–start problem is not common to the movie rating applications since there the data availability is higher, but could be ex- perienced in ITS use. For this reason it was crucial to have a partner with large data availability both on tasks and on students. We were able to select 100 students coming from the same school and that had already experience with the system, so that also the student cold–start problem could be avoided.

So, bkt initiation without data?

2/8

Back to teaching SImStudent Act 39\_3:)

The difficult part of this tutor is ‘coordinate in the problem’ to ‘the first row’.

How to restart server :

1. Close the step
2. Copy the link
3. Command+shift+delete \_ delete history
4. Paste the link.

My idea :

1. Chunk each coordinate (a,b) / (c,d) respectively
2. If student copy a (first number in the chunk/coordinate), he should copy c (first number in the other chunk). If he copy b, he should copy d.