AI agents for intelligent tutoring systems

Rohan D. Kekatpure

1. Problem description

We present design and implementation outline of an intelligent tutoring system optimized for 'teaching a new skill (basic geometry) to humans.' Our agent uses Frames-based knowledge representation but, being a complex system, integrates elements of other learning approaches.

The task at hand is to create an AI agent to help teach new skills to human in a classroom setting. Creating intelligent tutors is a topic of active research and there are attempts to create tutors for various skill subsets: writing, math, sciences and test prep. This broad and complex problem needs to be addressed iteratively by significantly restricting the problem scope (problem reduction).

After a brief consideration, one can see that the course materials help in two ways. First they offer a vocabulary to think about the problem. E.g. *whats the best knowledge representation for this task?*. Second, they offer problem solving strategies. E.g. *is this problem tractable via case-recording or would adaptation be needed as well*?

The presented design should be considered nothing more than a rough sketch. Up until now, we've thought about the course material exclusively in the context of solving RPMs. But we wanted to exercise our understanding by pondering a different problem. As such, the our presentation may not achieve the technical depth necessary for a full implementation spec.

2. What makes the problem difficult?

The difficulty of this problem can broadly be categorized into **educational** and **implementation** domains. The following series of (open-ended) questions capture the educational difficulties:

1. Any AI agent will need to measure progress in students' understanding of the concept being taught. Can mastery/understanding of a concept be quantified by a set of statistical measures?
2. Is the scope of intelligent systems restricted to teaching only basic skills or can we hope to graduate them to teaching skills that we consider 'advanced'?
3. Are intelligent systems restricted to teaching inductive logic (through use of production systems) or can they be used to teach deductive reasoning. E.g. can intelligent math tutors teach only algebraic manipulations or can they also teach skills in proof-type questions?
4. How can intelligent systems encourage alternative ways of problem solving?
5. Should our first attempts at building intelligent tutoring systems focus on semi-quantitative (e.g., more rules-based) fields like biology and chemistry rather than math, which require a creative application of a small set of axioms? Or should it be geared only toward test-prep (LSAT, MCAT, GMAT) which have a fairly well-defined set of concepts to be taught?

The above questions arise out of educational theory (and perhaps border on philosophy). To make progress the educational questions need not be answered fully, but only to the extent of allowing us scope our design iterations.

The implementation presents a separate set of questions that *must* be answered completely for an iteration to succeed.

1. What is the knowledge representation is appropriate?
2. What statistical measures will the agent use to quantify students' progress through the lesson (assuming statistical measures are appropriate) ?
3. How will the agent extract the information to be presented as hints? Will this also need to be pre-programmed or can this be generated from the knowledge representation?
4. How will the agent be trained to be adaptive and to enrich its knowledge-base?

3. Agent design and implementation outline

At this point, we need to narrow the scope and select learning category to illustrate the design of our agent. To keep things simple, we'll assume that we're designing an agent to teach basic geometry of circles — specifically the inter-relationships between radius, diameter, area and circumference.

3.1. Knowledge representation

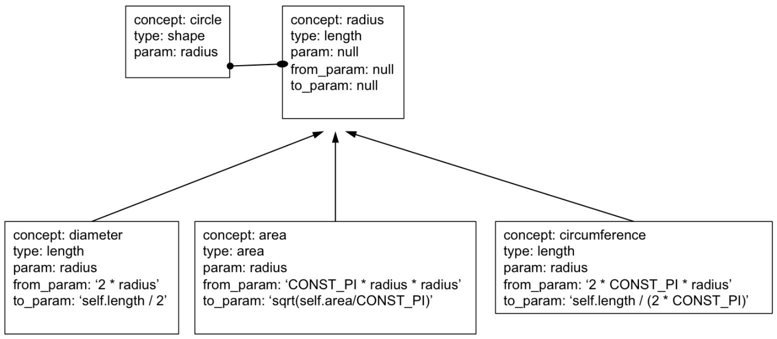

Fig. 1. Representation of circle concepts as frames.

With some reflection, it is apparent that majority of the design for this agent involves choosing a proper knowledge representation and echoes Prof. Goel's statement earlier in the course. Presently, we choose **frames** as our knowledge representation tool. Frames are chosen to represent the concept, inter-relationships, question profiles and student profiles. One possible frame representation of circle concepts is presented in figure 1.

In code, frames can be implemented in memory (using dictionaries/hashmaps) in files (using JSON/XML serialization) or in properly normalized databases tables.

Given this representation, we can construct a question template that'll allow us generate questions at various levels of difficulty. These graded questions will help the student learn about various geometric quantities and their computations from each other (area $\leftrightarrow$ radius, area $\leftrightarrow$ circumference etc). One possible question template is shown below:

```
target: ${target}
given: ${given}
value: ${val}
```

3.2. *Question generation*

About a dozen concrete frames can be generated from this template. A simple Python snippet below will illustrate our point:

```
targets = "diameter circumference area radius".split()
givens = "radius diameter circumference area".split()

for given in givens:
    for target in targets:
        if target != given:
            print "What is the {} of a circle with {} = 10?"\
            .format(target, given)
```

This results in the generated questions:

```
1. What is the diameter of a circle with radius = 10?
2. What is the circumference of a circle with radius = 10?
3. What is the area of a circle with radius = 10?
4. What is the circumference of a circle with diameter = 10?
5. What is the area of a circle with diameter = 10?
6. What is the radius of a circle with diameter = 10?
7. What is the diameter of a circle with circumference = 10?
8. What is the area of a circle with circumference = 10?
9. What is the radius of a circle with circumference = 10?
10. What is the diameter of a circle with area = 10?
11. What is the circumference of a circle with area = 10?
12. What is the radius of a circle with area = 10?
```

3.3. *Hints*

Hints and leading questions are crucial for learning and our frames-based knowledge representation allows us to generate hints in a straightforward manner. Following the principle of Socratic learning, hints will be provided in the form of leading questions as illustrated by the hint templates below. Each can be made concrete by populating the `${variables}` with values from the concrete question frame used to generate the question.

```
1. According to the lesson what is the parameter for a circle?

2. Is ${given} a radius?
```

If the answer is ‘yes’ skip to question 5. If ‘no’ follow up with additional questions:

```
3. What is the formula for radius in terms of ${given} ?

4. Ok, can we now get radius from ${given}?

5. What is the formula for ${target} in terms of the radius?

6. Ok, can we now get ${target} from the radius?
```

From the knowledge representation in figure 1, the agent itself is aware of the answers to the hint questions. For hint-question 1, it knows that the fundamental parameterization of the circle is in terms of the radius (and center, which we have ignored here). For hint-question 3 and 5, the knowledge representation (`to_param` and `from_param`) supplies formulas for the calculation of targets from radii and back.

Lets imagine a student working through question 8. Recall that this question is generated from the frame:

```
target: area
given: circumference
value: 10
```

Further imagine that the student needs all possible hints. The simulated interaction would proceed as follows (statements beginning with $ = agent, >> = student):

```
$ According to the lesson, what is the parameter for a circle?
>> radius

$ Is circumference a radius?
>> no

$ What is the formula for radius in terms of circumference?
>> radius = C / (2 * PI)

# If the student doesn't know advice going back to the lesson
# or provide formula from the knowledge representation. This
# behavior can be controlled by a setting.

$ Ok, can we get the radius from the circumference?
>> yes

$ What is the formula for area in terms of the radius?
>> A = PI * radius * radius

$ Ok, can we now get the area from the radius?
>> yes
```

3.4. Measurement of student progress

Student progress can be measured by assigning difficulty points to each question and maintaining a running total of points. The agent can also record the pace of each student (how fast the questions at each level of difficulty are being solved), the number of hints required, rate of success on a question after 1, 2, 3, 4 . . . hints. All of this information can be stored in a `Student` frame.

The progress information in the ‘circles’ section can be merged with overall student profile to create adaptive lesson plans.

3.5. Agent training

There are two aspects to training the agent for intelligent systems.

1. Training the agent to be adaptive to the student needs
2. Training the agent to enhance its knowledge-base

3.5.1. Agent training for adaptability

The agent can learn adaptability by constantly modifying the student and question profiles. In section § 3.4, we outlined how student progress is monitored and added to the student profile. The same session data can be used to **enrich** the `Question` frame and to build a profile of a question over time. The agent can record the completion time, mean number of hints, success rate after hints etc. After a period of time (i.e. after a few thousand served questions), this accumulated data will be used to train the agents' response.

In a combined **case-based + production systems** approach, the enriched data will be used as features to compute a 'distance metric' between the question serving contexts. The distance metric will them be subjected to a set of rules to generate the learning experience. For example, on a given question, if progression of the current student matches one historically, the agent can anticipate the hints that the student is likely to need and can highlight the relevant sections from the text to brush up before commencing or generate formula cheat sheets (from knowledge representation in Fig. 1). An example of an enriched `Question` frame is provided below:

```
target: area
given: circumference
value: 10
times_served: 500
times_correct: 125
success_rate: 0.25
avg_hints: 4
time_reqd: 600
```

3.5.2. Agent training for knowledge enhancement

An agent trained to enhance its own knowledge base will be capable of performing modifications to its existing concept frames (Fig. 1). Perhaps more importantly, such an agent will also be able to create **new** frames of its own.

So far, we have assumed that the agent was operating based on bootstrapped, pre-programmed frames. These frames described certain basic geometric concepts and their first-level connections: circles (radius, diameter, circum, area), squares (side, diagonal, perimeter, area), triangle(sides, perimeter, area), and so on.

Warning: This section may make your head explode due to the amount of speculation.

An agent capable of creating knowledge, it will need a scalable mechanism to consume human-language input. One input source can be digitized geometry texts. The agent will consume paragraphs and generate frames for new concepts. This would necessarily mean that the agent needs to be integrated with natural language processing (NLP) modules. The basic premise of this module is simple: consume definitions of concepts and output frame. For example, consider the description of a circumcircle of a square from hotmath.com:

```
When a square is circumscribed by a circle, the
diagonal of the square is equal to the diameter of
the circle.
```

The NLP module will need to parse this and generate a new frame as follows:

```
concept: circumcircle
type: circle
contains: square
param: radius
raw_relation: 'square.diagonal = circle.diameter'
to_param: 'self.radius = self.square.side / CONST_SQRT2'
from_param:'self.square.side = self.radius * CONST_SQRT2'
```

The trick is in deriving the `from_param` and `to_param` from the `raw_relation`. The constants like `CONST_SQRT2` appear from the relationship of the side of the square with its diagonal.

Once the circumcircle concept is successfully created, it is a relatively easy task to generate test questions and hints using question frame templates as detailed in sections § 3.2 and § 3.3.

Note that the NLP module is a requirement for scalability of concept generation, not knowledge enhancement itself. The system outlined so far is capable of enhancing its knowledge if more and more frames are supplied. For example, one can certainly imagine generating massive knowledge bases via crowdsourcing.

Conclusion

We presented the educational and implementation challenges in creating AI agents to power intelligent tutoring systems. Following that, we provided a rough design and implementation outline of an AI agent that could be used as a basic geometry tutor. The agent has inbuilt knowledge, represented as frames, of basic geometrical shapes as well as template frames for questions. The agent is able use the question frames to generate questions as well as hints. We also presented ways in which the agent could use the session data to build profiles of a students and questions. After some accumulation period, the profile data is used to create train the agent for creating adaptive learning experiences based on the question difficulty and the student ability.

In its current iteration, the system is limited by the number of bootstrapped frames. Enhancing the agent's knowledge base will require feeding it concepts from human-language text sources(will require integration with text-understanding modules) or crowdsourcing frame generation. Once the frames are generated, however, the existing question and hint generation process can consume them to create learning experiences.