

An artificial intelligence experiment in college math education

Oliver Knill, Johnny Carlsson, Andrew Chi and Mark Lezama

11/17/2003, latest edit: 4/19/2004

Abstract

To gain more insight into the process of teaching and learning mathematics, we experimented during the Fall of 2003 and the spring of 2004 with some calculus chat bots. These bots know an encyclopedic glossary of mathematics definitions as well as general knowledge background and can solve simple mathematical template problems posed in colloquial language. The bot "Sofia" can also interact with computer algebra systems like Pari or Mathematica to solve simple pre-wired calculus problems.

1 Introduction

1.1 Artificial intelligence in education.

Artificial intelligent projects like "Eliza" [8], "Student" or "GPS" [5] attempted to produce agents which can solve general mathematical problems. One soon realized that this goal was too ambitious. Not only because of technical hurdles, but also because one had to realize that many problems are computationally hard and can not be settled in reasonable time. While this slowed the momentum of AI developments in education, it did not stop it.

1.2 Thinking by Searching.

A branch of artificial intelligence has since focussed on building sophisticated data structures, using a large knowledge database. The difficult and complex task of finding a problem is reduced to "used knowledge" ([2] p. 74: "The most efficient way to solve a problem is to already know how to solve it. Then one can avoid search entirely. Accordingly, a branch of AI research has sought to find ways to embody knowledge to machines"). Students learning mathematics often use this strategy also: knowing how to tackle basic prototype problems (and most of the calculus textbook problems are like this: compute the derivative of that, find the line integral of that etc)), allows to pass a typical calculus course reasonably well. We know that, since nonstandard problems are usually solved catastrophically badly, even at Harvard.

1.3 Search engines and AI.

Also popular search engines illustrate the power of this AI branch: the web as a database becomes - when searched and mined - a proper "intelligent agent". Indeed, modern search engines can "answer questions" surprisingly well, if they are of lexicographic type. Good search engines understand questions formulated in colloquial language. The huge amount of information and the sometimes not predictable search outcome adds a creative element to the answer - something which is often a priory thought impossible of intelligent agents. The search engines already realize the vision of a "semantic web", without additional effort of the information providers, like the individual web-pages. This is good so, because one does not want to burden the information sources with complexity:

if the process of making information available is complex, it is not done.
--

2 The goal of the project

2.1 The goal.

The goal of our project was to gain insight into the mechanisms of teaching and learning by playing and experimenting with some existing AI tools. We hoped that we could use the experiments to shed light on basic questions in education like:

- How do we teach effectively?
- How do students learn?

We hoped also to produce "agents" which are of practical value. The knowledge base can be used to produce glossaries or help to build calculus problems of a certain type. Our goal were to

- Build an online partner for mathematical conversations.
- Have an interface to computer algebra systems.
- Learn about artificial intelligence. What is possible and what not?
- Explore new features in mathematical communication.
- Understand the teaching process from an other perspective.
- Reflect on the process, how we learn mathematics.

2.2 What is a chat bot? The interaction of humans with a computer can happen in different ways. We can read documents, search and interact with a computer via terminal or browser, use computer algebra systems or build our own tools using any programming language. A different type of interaction is realized by an "intelligent agent". It can be used as an interface to other tools or agents. An intelligent agent should by definition contain a

- knowledge base
- a problem solving component
- an interface to the user as well as
- a knowledge acquisition and
- a learning component.

[7].

2.3 Comparison with other information containers.

We can compare the intelligent agent with other information containers or processors:

Type	history	programmable	robust syntax	can learn	extendable
Database	no	yes	no	no	yes
Document	no	no	yes	no	no
Terminal	yes	yes	no	no	yes
CAS	yes	yes	no	no	yes
IA	yes	yes	yes	yes	yes

2.4 The problem with databases.

A database gives localized answers to localized problems. They have their uses, but they can be frustrating:

- Entering a slightly wrong search produces no result. One can try this out in most online news services, where the search is frustrating. An other example: have you ever got a list of all libraries at Harvard using existing databases?
- Databases hide their internal structure. The can not be audited easily about their size, their correctness etc. Example: Wikipedia, the online Encyclopedia. It becomes searchable because the website is online and can be minded. Search engines, which have increasingly good AI capabilities help to sort out the relevant data.

3 Learning mathematics

3.1 Understanding text.

The problem to distinguish between unsolicited email and regular email illustrates the struggle of artificial intelligence to understand and classify written text. One reason for this difficulty is the presence of **context** in conversations. The context can come from common experiences or previous conversations. A frequent problem, a teacher or seminar speaker faces is that the written information might well make sense to him or an expert, but not to the audience. This is especially in the process of learning mathematics, where we have to get acquainted with an abstract formal framework.

3.2 A simplistic model for understanding Mathematics.

A simple model for understanding a mathematical concept is to split the learning process into four levels: What?, How?, Why? and Why not? These four levels are:

- 1) Know the definitions
- 2) Master the algorithms
- 3) Understand the concepts
- 4) Be innovative

Each concept builds onto the next. For example, it is hard to go through an algorithm, if the objects are not known. Computing the flux of a vector field through a surface can not be done, if the definition of parametrized surfaces, the definition of the cross and dot products etc are not known. It is difficult to understand concepts, if one can not apply it. For example, it is difficult to understand, why a non-symmetric matrix can in general not be diagonalized, if one does not know the process of diagonalization in specific examples. The conceptional understanding needs the algorithmic understanding. It is also unlikely to be creative, if the concepts are not understood. For example, it would be difficult to come up with the Jordan normal form theorem if one does not deeply understand diagonalization and its limits. The last step, the innovation, is the most complex one and we did not attempt to look at it: it is a fascinating subject even when specialized to Mathematics. It is treated in classical books like [1, 4, 9, 3]. The history of mathematics provides many tales on such achievement has been done.

3.3 Examples

Lets look at some examples on these four planes of understanding in mathematical context:

- "What is the definition of a gradient?" The task is to look up the definition $\nabla f(x, y) = (f_x(x, y), f_y(x, y))$. Students can memorize the answer. It is also no problem for a machine to answer that.
- Lets look at the task: "Extremize $f(x, y) = x^2 - y^2$ under the constraint $x^3 - y^3 = 1$ ". It is an algorithm. One has to solve the system of equations $\nabla f = \lambda \nabla g, g = c$. The algorithm is straightforward. The only problems students of calculus face here is the complexity of solving the nonlinear system of equations.
- "The gradient is large if level curves are close to each other." This is a conceptual problem in which the student has to understand what the **meaning** of the gradient is. An intelligent agent would have to know: the size of gradient is large if level curves $f(x, y) = k\epsilon$ are close without having memorized that fact. Exam problems which involve this type of conceptional part are often solved poorly.
- Lets look at the problem: "Find examples, where a problem with constraints can not be solved by substitution, but can be solved with the method of Lagrange multipliers." This is an assignment which requires some creativity. In general, a Lagrange problem can be solved by plugging in $y = h(x)$ into $f(x, h(x))$ and obtain a function of one variable which one can extremize. This approach fails, if one can not find a closed formula for y . The task to find an example, where it is still possible to solve the Lagrange equations but not the substitution part. This needs some thought and the ability to play around.

As with any model, this "what-how-why-why-not" model is a simplification. The real world is much more complex. For example, different levels are linked to a certain degree.

- Understanding helps to learn the definitions and algorithms, but certain "understanding topics" can become definitions.
- There are cases, where it makes sense to start with an innovative approach first. A good hook-up question ("Hatsumon") can be fruitful to motivate a story, even if at the beginning, one does not see, how things work.
- Not knowing all details can actually help to find new things, solutions which seem crazy for someone who knows the subject.
- Sometimes, it is possible to master an algorithm, even so the object is not known. For example: most people apply the commutativity rule $x + y = y + x$ correctly, without knowing the name of the rule.
- Sometimes, it is easier to master an algorithm if the concepts are understood. There are cases, where mastering the algorithm even requires a deeper understanding of the concepts. This is especially true for algorithms which are nonlinear. This happens in calculus when treating series or solving nonlinear systems of equations in the topic of constraint optimization. Other problems which require some imagination is the decision whether a series is finite or infinite and where sometimes majorization or minorization is needed.

3.4 Understanding Mathematics.

A machine can deal well with **definitions** (what) and **algorithms** (how). More problematic are the realms of **understanding** (why) as well as the **innovation** level (why not). It is general challenge in the process of teaching to solve the "meta problem" and to teach solving problems, enable so a student to solve problems which are not templates. This is the holy grail of education as well as AI.

3.5 How do we learn?

There are different ways, how we can learn a new subject: we read, talk, write, think, solve problems, experiment, teach, play or even program it.

Minski [2] expresses the idea of teaching as "building a robust network" in the brain.

Because this is a complex process, talking to an expert is probably the most efficient way to learn a subject fast. The problem with many college curricula are that the student-teacher ratio is so large that it is difficult for every student can have regular discussions with the teacher.

4 Enriching the teaching spectrum

4.1 Mathematics activities.

Learning mathematics is a complex process. Because different type of learning types exist, teaching succeeds best if a variety of options are offered:

- lectures
- projects
- labs
- discussions
- web-based tools or
- homework

Experience shows that teaching works best, if a wide spectrum of activities and opportunities are offered.

4.2. Lowering barrier to discuss mathematics Chat tools can be compared with a online forum. Online discussion forums in mathematics are useful only if they are well moderated and also then, we experienced that students are often reluctant to share their mathematical struggles with others online. Unlike in social sciences, where answers are not always final and different opinions can coexist, being wrong in a mathematical statement can be embarrassing when published and stored in an online forum. A chat robot is more forgiving, students talk anonymously, they can ask and discuss more freely.

4.2. A new building block.

An AI agent can enrich the teaching environment curriculum by adding a building block to this menu. It is a friendly interface to a knowledge database. Unlike a usual database which is in general limited and picky, an AI agent accepts arbitrary input and can answer questions posted in colloquial language. This requires abilities which go beyond a relational database. The later can get stuck for example if data are not entered in a specific form or have mistyped words. A commonly used and constantly improving chat bot is google. One asks a question, the answer is returned as a list of links pointing to answers.

The mark-up language AIML [6] provides a framework for such a chat interface. While existing robots are still quite limited and by far fail the Turing test, the results are sometimes surprising. A chat robot does not replace an ordinary database, but it can enhance existing databases, can be used as an interface.

4.3. Watching students learn An AI agent can enhance or change the teaching and learning experience. Having the tool answering questions could help the teacher to see, where students have problems, it could stimulate the instructor to understand how students learn. It could improve and enhance the overall quality and especially variety of teaching. It could also be used as a scientific tool to gauge student learning. Last at not least, it can be fun to play with it.

By accessing logs, we can see how people learn, what questions are asked. The logs can provide information how useful the tool will be for students. It helps to improve the database.

Having data on the interaction with users is the main advantage over a simple database. Of course, reading and evaluating these data uses resources. Serious data mining would be a project by itself.

4.4. The local calculus teaching culture.

At the Harvard mathematics department, where calculus is taught mostly in small sections, students are offered a wide spectrum of activities like lectures in classes, sometimes with demonstrations, slides or enhanced by classroom response systems, homework assignments. Besides reading in textbooks, working on computer-based quizzes or online activities or computer labs students can also participate in problem sessions which are run by course assistants as well as meet with instructors during office hours. This is a rich learning environment already. Do we need more? We believe that an online database in form of an AI chat bot for Calculus could enrich the learning experience and add an other channel for learning mathematics. It could be useful to look up information about mathematics and other disciplines.

4.5. The danger of a teaching mono-culture.

It is important that any enhancement of the teaching should primarily **complement** existing instruction tools and not **replace** them. Thinking about how a machine learns and interacts can sharpen the teachers awareness about the process of teaching and learning of mathematics. The tool is therefore also a scientific experiment. Since we have never seen such a tool used for teaching math, we want to find out, whether it is useful.

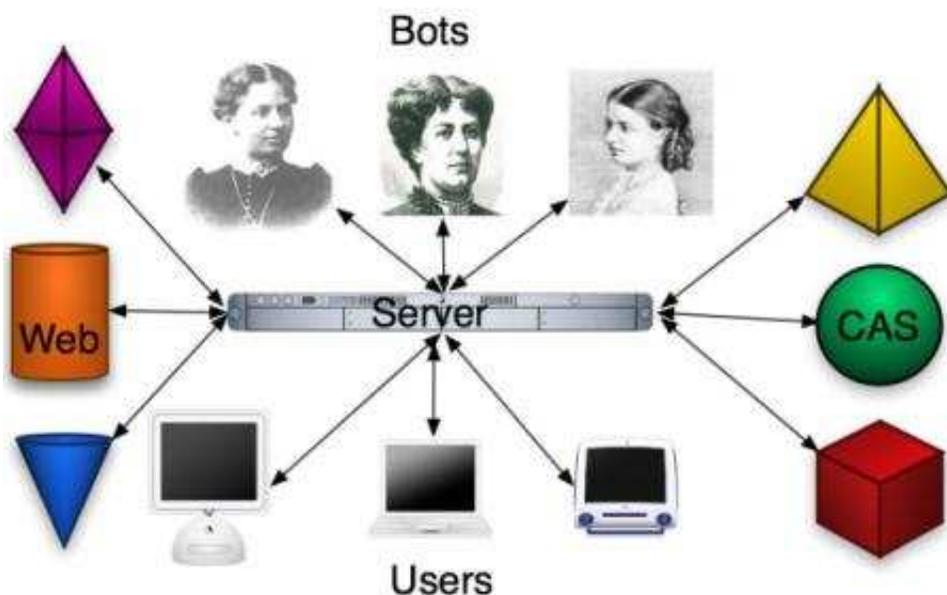
5 Sofia

5.1. Reusing existing building blocks.

We first worked with existing AI components and embedded them into a larger "wrapper". This allowed us to use different components simultaneously, have bots chat with each other and overcome limitations when bots interact with computer algebra systems. Especially the machine-machine interaction which is easy to implement, can be enlightening.

Treating AI agents as "black boxes" or modules allowed us to exchange them or have several different components work together. Having independent tools work together to build a larger tool is at the heart of the Unix philosophy and is an efficient way to build and debug projects.

In order to be more flexible, we then built a web interface, which can interact with the web, with more bots as well as with computer algebra systems. Additionally, this interface will handle an additional "context channel" which reflects information exchanged in earlier talk. The interface makes it also possible to learn things.



5.2. A fresh implementation.

We also built from scratch our own implementation of a bot. The limitations of existing bots are:

- The memory is limited to the memory available on the computer. The bots usually load everything into the memory.
- The program is not scalable.
- The tools are programming language dependent (Java, C)
- They are difficult to extend.
- The tools are error prone. If a part fails, all fails.

We take a sentence, proof-read it, simplify it and then translate it into a path of a Unix file system. This path leads to a program, which is then executed. The individual programs are the nodes of the neural net. Learning means building new paths, new nodes or (not yet implemented), distributing the weights on which path to chose.

Part of the Unix file tree is generated by hand, part of it (like for example definitions) is generated by the machine from simple text files. The nodes of the tree can be any programs we use CAS systems, programs which harvest the web, perl programs or simple shell scripts. The bot itself consists of a few lines of codes only. The brain itself can be entire unix file system. Using such a "database" minimizes the search task, is extremely reliable and not limited in space (the only limit is the harddrive space available). An other advantage is that different people can work on different subtrees.

5.3. Building an online Encyclopedia.

The brain of the bot is a collection of files. We enter the information as plain text files. Perl scripts translate the information into the form, the bots can understand.

While we focus on mathematical content, we have to build up "common local knowledge" which will make the robot more friendly. For example, the robot should know some math jokes, be familiar with movies, in which mathematics plays a role and be acquainted with local college knowledge.

As a side benefit, the tool can provide an open source database for calculus and other parts of Mathematics. This database is text based and contains the definitions as well as prototype examples. The resources can be used together with other multimedia educational resources like Merlot and become part of a public domain mathematics encyclopedia like Wikipedia or Mathsource. Any reasonable online database of knowledge of a certain size can only be realized using the common effort of many people.

Any reasonable online database of knowledge of a certain size can only be realized using the synergetic effort of many people. Having the information database as well as the conversion and data mining tools open source will allow others to build on these resources and improve them.

From the AI point of view, we only aim at a lowest level of language understanding ([10]). We are building a **parse tree** which leads to a response computed by an arbitrary program located at that node. The world model to find meaning in the structure of the sentence is confined to a predefined class of structures.

The reduction of the sentence is given by **context free rules** which are integrated in the form of filters.

5.4 Interaction with computer algebra systems.

A part of Mathematical thinking is to do calculations. Computer algebra systems can do that too. It is natural to let an AI agent access such a CAS device in order to enhance the answers.

The AI agents we used first do not digest well strings entered to computer algebra systems. This forced us to enter strings coded as common words. The bot would then treat such words in the same way as colloquial language.

We have experimented with

- Mathematica
- Pari and
- Maxima

These CAS allow an interface via shell.

5.5. Implementing Context

Language happens through different channels and not only through speech. If a teacher talks to a student, she can draw on paper or the blackboard, enhance certain things with body language, eye-contact or humor. The conversation usually has a context given by past conversations, commonly known events etc.

A main weakness of AI agents is that context is missing. Some of this deficit can not be made up at all. Here are some context channels:

- past common experiences,
- body and language,
- or the talk itself.

Context in communication

We see already with email communication, how difficult it is to explain things. Even telephone conversations are trickier than face-to-face talks because visual components are missing.

Existing bots take some context into the picture by sharing information using "cookies". The bot, you are talking to knows the users name, etc.

We experimented with techniques to pass meta knowledge around. The simplest is to submit forth and back a history of the speech. A sentence in the middle of a speech is often unintelligible without the past. Example:

Student: Is the function $\sin(x)$ continuous?

Bot: yes, it is. Every trigonometric polynomial is continuous.

Student: Is it differentiable?

Bot: What?

The bot has to know the past conversation in order to continue. The question "Is it differentiable?" does by itself not make sense.

Technology: we can send forth and back context information as hidden variables on the web. The user does not see this information. The bot can access it. We were able to implement this but there remains the problem to extract from the context in order to have the bot looking things up in the right place.

5.6. Learning

Within the current scope of the "Sofia" project, we define "learning" to be the acquisition and encoding of new knowledge. We were particularly interested in achieving automated interactive learning, but there are other aspects.

- The most straightforward way to make Sofia learn something, such as the definition of a mathematical term, is to **hardwire it into her brain**. An example of an entry is:
[how are you] I'm fine.

We added a lot of hardwired answers in mathematical context.

- AI agents **log their talk**. This can be used in principle to feed back information into the brain of the bot. The "targeting tool" provided with the A.L.I.C.E. software helps with that. In our view, this approach does not allow Sofia to learn in a timely manner and it relies too almost entirely on the bot-master to do the teaching.
- We made learning more effective, by introducing it into the language. By telling the bot to learn, we can force feeding the entry into the brain.
sofia "learn [you sound tired] I worked all night"
- By letting bots talk to each other, it is possible to let them teach each other.
- Finally, the bots could look up things on the web.

The problems of learning are

- Filtering out garbage and abuse.
- Know the reliability of the data.
- Avoiding duplication.

The problems of filtering data and estimate their reliability is important and an issue for many data collecting projects on the internet. For example, the online Wikipedia (www.wikipedia.org) is an ambitious data collecting tool. Many authors feed the "Wiki" with articles and information. Everybody can be an author. Therefore, unimportant details can become important, errors are possible. The collaborative work and rework seems to increase the reliability of the data over time. The reader is the peer reviewer as well as the author. It is clear however that such data have to be taken with more caution than articles written by proven experts in a field.

5.7. Examples of interactions.

```
sofia "how are you"
sofia "compute the derivative of sin(10*x)"
sofia "find the critical points under constraint of 13-4*x+x^2-6*y+y^2+z^2 and z^2-x^2-2*y^2"
sofia "find lagrange solution x^2+y^2+y and 4x-5x^2-y^2-3*y-3"
sofia "what is a determinant"
sofia "add three and four and seven"
sofia "add million thousand hundred fifteen"
sofia "compute PrimeQ[33]"
sofia "compute Det[{{3,4,1},{32,4,1},{1,2,3}}]"
sofia "what is an integral"
sofia "compute the number of partitions of 10"
sofia "what is the weather"
sofia "learn [Switzerland] Country in the center of europe."
```

6 Artificial intelligence in education

6.1. The Turing test in education.

Can a machine solve routine calculus problems formulated in common language? Can it solve homework problems in a typical calculus curriculum or pass a placement test in mathematics? Can a machine get a degree from a university?

6.2. Can a machine help to produce exams or homework problems?

Yes, a machine can assist with the production of exam and homework problems mostly in conjunction with a computer algebra system. Some educators do that in order to generate better problems.

Example: to produce an integration problems of the type "Find the anti-derivative $f(x)2x/(1+x)^3$ ". the machine produces a random function like $f(x) = 2x/(1+x)$ and computes $f(x)f'(x)$, knowing that its anti-derivative is $f(x)^2 = x^2/(1+x)^2$. The solution path which the student will go like using substitution $1+x = u$ is different from the path the machine has taken to produce the problem.

We suspect that many problems in calculus text books have been produced like this and that many teachers have built their own "problem generation software". It is not difficult to build such tools with current computer algebra systems like Mathematica.

Additionally a bot can also be used to look up problems in databases in order to produce intelligent problems. Most calculus problems appearing in introductory calculus series books follow a pattern of maybe a few hundred problem cases. Knowing a few dozen problem types allows a student to pass the exam.

6.3. Replacing educators by machines. Will machines hold office hours, answer student questions or even teach classes in the future? The challenges in distance learning show that the social aspect of learning is very important and that robots teaching a classroom will be very unlikely. Experiments with "ubiquitous computing" have shown that too much computing is not successful if the addition of computers went to the cost of teacher salaries.

The teacher is the backbone in the teaching process. Technology like Computer algebra systems, multimedia presentations or "chat bots" can serve as amplifiers but not replace a good guide.

We believe however that artificial intelligence can enhance the teaching process like other technological media: from computer algebra systems, online discussions or multimedia. It is a tool which has not been explored very much, but which adds to the variety of teaching tools.

7 Conclusion

- Teaching works best when the material is offered in many different ways.
- Allowing students to interact with a calculus chat bot adds to the variety of tools.
- Building a knowledge database can help to broaden the publicly available information pool.
- By watching a machine learn Mathematics, we can learn how students learn mathematics.
- By teaching a machine, we get insight into the process of teach mathematics to humans too.

Acknowledgments. This project was supported by the Harvard Provost's fund for Innovation in Instructional Technology.

References

- [1] J. Hadamard. *The Mathematician's Mind*. Princeton University Press, Princeton, New Jersey, 1945.
- [2] Marvin Minsky. *The Society of Mind*. A Touchstone Book, published by Simon @ Shuster Inc, New York, London, Toronto, Sydney and Tokyo, 1988.
- [3] D. Perkins. *The Eureka Effect*. WW. Norton, New York, London, 2000.
- [4] G. Polya. *How to Solve It*. Princeton University Press, Princeton and Oxford, 1945.
- [5] S. Russell and P. Norvig. *Artificial Intelligence: A modern approach*. Prentice Hall, 1995, 2003. second edition.
- [6] A.Taylor J.Baer D. Daniels T. Ringate, R. Wallace. Aimi primer. 2002. www.aitools.org.
- [7] G. Tecuci. *Building Intelligent Agents*. Academic Press, San Diego, London, Boston, New York, Sydney, Tokyo, Toronto, 1998.
- [8] J. Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9:36–45, 1965.
- [9] W. Wickelgren. *How to Solve Mathematical Problems*. Dover Publications, Inc, New York, 1974.
- [10] P.H. Winston. *Artificial intelligence, second edition*. Addison-Wesley, Reading, Massachusetts, 1984.