Automating “Human Like” Example use in Mathematics

Alison Pease, Daniel Winterstein, Irina Preda,
Paulius Kuzmickas, Ursula Martin

Overview of interests

RQ1: How do people do mathematics?

- What do they talk about?
- How do they explain things?
- What do they value?
- What are the patterns of communication?

RQ2: What is the role of the machine in this?

- To enable communication?
- To perform some of the “drudge” tasks?
- To provide new perspectives on theories about human-produced mathematics?
- To contribute in a creative way to the production of mathematics? How can we build collaborative systems? Could an autonomous machine usefully contribute to mathematical discussion?

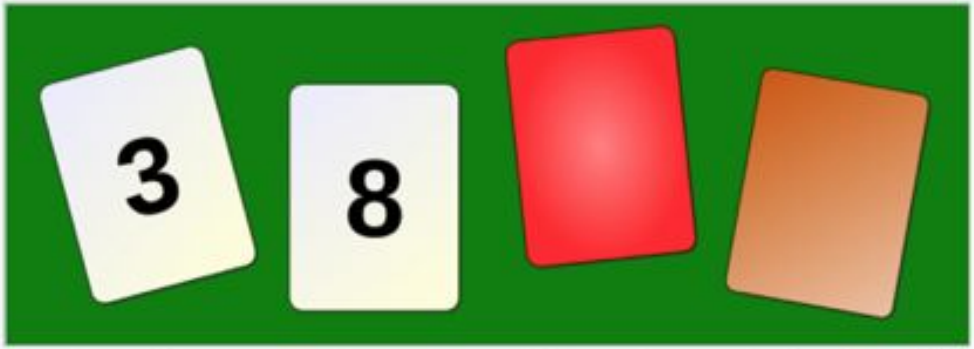
Automating Human-like Example use

RQ1: How/when do people use examples in collaborative mathematics?

RQ2: Could a machine introduce an appropriate example at an appropriate time in an appropriate way?

What is an example?

- Examples of a **concept**, such as the set of natural numbers being an example of a *group*, and the numbers *3*, *4*, and *5* an example of a *Pythagorean triple*
- Supporting or counterexamples to a **conjecture**, such as *2* and *3* being supporting examples of the conjecture that *all integers have an even number of divisors*, and *4* being a counterexample.



Each card has a number on one side, and a patch of color on the other. Which card or cards must be turned over to test the idea that if a card shows an even number on one face, then its opposite face is red?

What is an example?

- Examples of a **concept**, such as the set of natural numbers being an example of a *group*, and the numbers *3*, *4*, and *5* an example of a *Pythagorean triple*
- Supporting or counterexamples to a **conjecture**, such as *2* and *3* being supporting examples of the conjecture that *all integers have an even number of divisors*, and *4* being a counterexample.

Why examples?

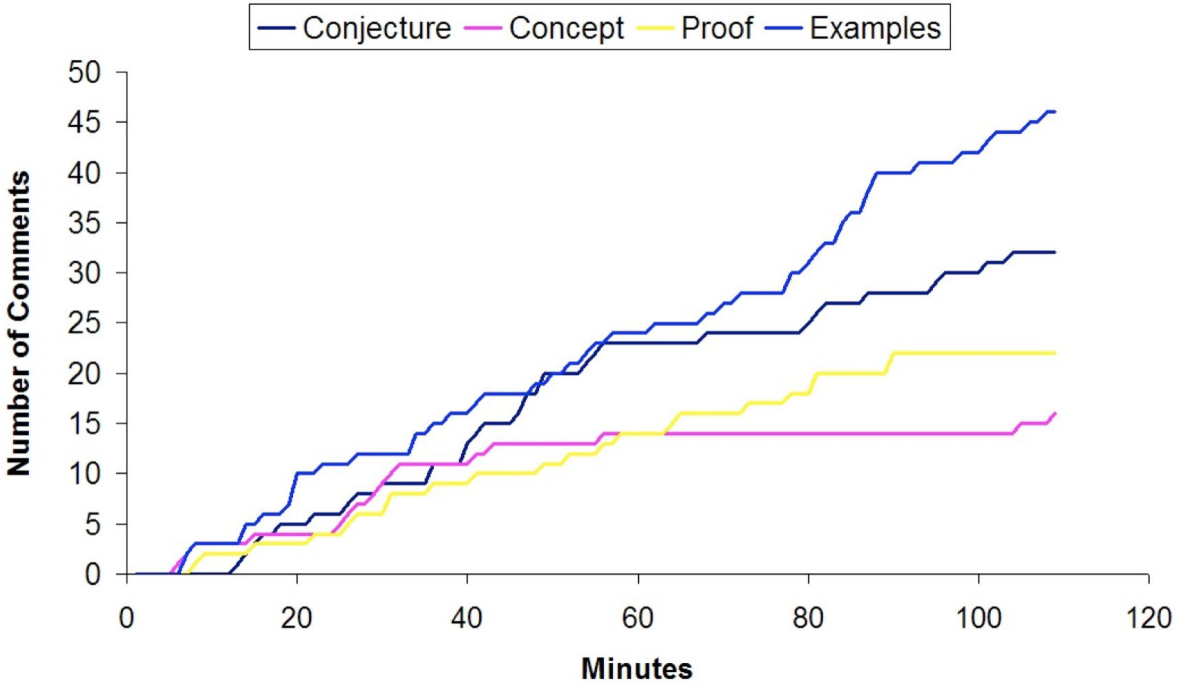
We conducted a course-grained analysis of Question 2 of the 2011 IMO to develop a typology of comments (solved in 74 minutes by 27 participants through 174 comments on 27 comment threads):

Concepts: You could define “the wheel of p”… (10%)
Examples: If the points form a convex polygon, it is easy. (33%)
Conjectures: One can start with any point (20%)
Proof: Maybe the strategy should be to take out the convex hull of S from consideration; follow it up by induction…(14%)

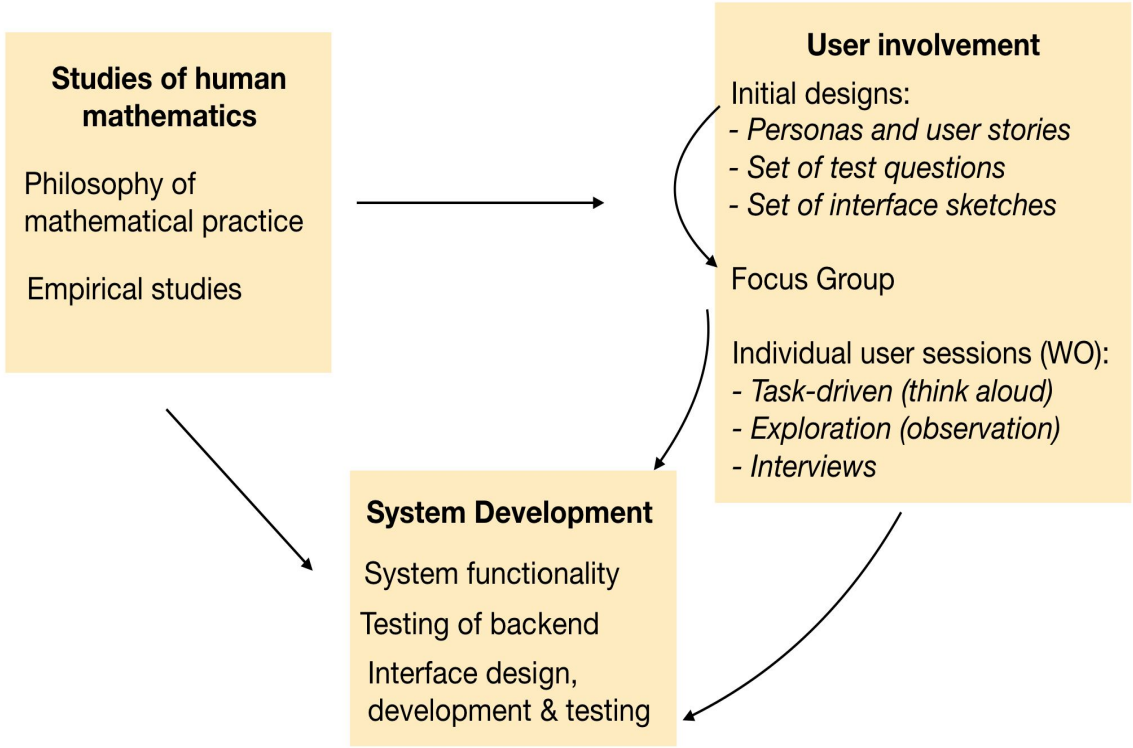
Other: I think that is a good start, thanks Varun! (23%)

In another study of a sample MathOverFlow conversations we found that in a third of the responses explicit examples were given, as evidence for, or counterexamples to, conjectures.

Why examples?



Our approach: Three strands



Document: mpm1-2009

Line #'s | Memos | RTL — Added: 12/04/2013 Creator: alisonp Excerpts: 559 — Memos: 0 — Descriptors: 0

gowers

54 60. Another small case. Let's take a_i=i for i=1,2,3,4. So we're trying to get to 10 in steps of 1,2,3,4 and there are three landmines.

If there's a landmine on any of 1,2,3,4, then by 47 (@liuxiaochuan) they must be on 4, or 4 and 3, or 4 and 3 and 2. In the third case we can go to 1 and then to 5, and then we're done by induction (two steps and zero obstacles, so perhaps induction was a bit of a sledgehammer). If there are obstacles on 4 and 3, then induction is more appropriate — we can either get to 5 in two steps and are then done, or there's an obstacle at 5, in which case we can go 2,6,7,10. If there's just an obstacle at 4, things get harder, since then we need to know what goes on after 4. But then we can cheat and say that at least one number between 6 and 9 is an obstacle so we can run things in reverse. The only case not covered is then when the obstacles are at 4,5,6.

That was still a rather ugly case-by-case argument, but it serves to confirm a sense that the difficult case is when the obstacles are not near the end points.

I'd like to try to find an argument along the following lines. Order the step sizes as a_1<\dots<a_n. Now let's try two paths. The first is where you take the steps in increasing order of size, and the second is where you take it in decreasing order. Now look at where you are half way through this process. Suppose that in the first case you have passed well under half the obstacles and in the second case you've passed well over half. Then it should be possible to move from one extreme to the other and find a permutation where you've passed more or less exactly half. (Actually, of course, the hypothesis here doesn't have to hold, but this is just meant to give the flavour of some kind of argument.) And then there might be a hope of… hmm … I'm still trying to find that elusive jump over two obstacles that takes place at exactly the right time.

Subquestion. If a_i=i and you have n-1 consecutive obstacles, what's the neatest proof that you must be able to get to the last non-obstacle without using a_n? (I don't think it's hard to prove it, but it would be good to have something that had a hope of generalizing.)

1

0

Rate This

20 July, 2009 at 12:03 pm

gowers

63. Re 54. Your analysis shows that it is possible for the gaps all to be at least a_n-1. Just let all the a_n be very large and roughly equal to a. Then the sum is approximately equal to na, so if we start near 0 and end near na, then we can get gaps of size about na/(n-1), which is bigger than a_n. So my proposal runs into difficulty. Not sure how big a difficulty it is though.

1

0

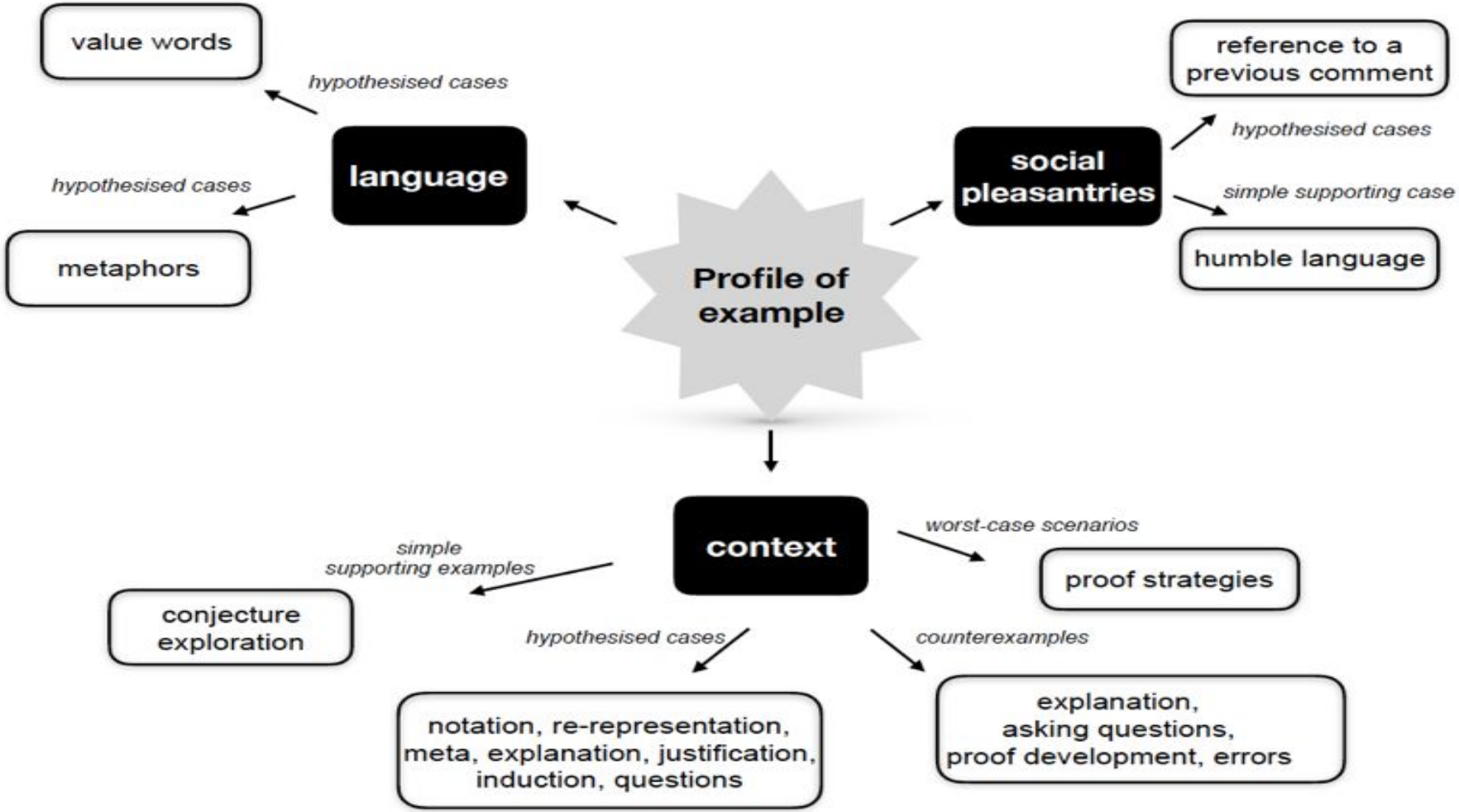
Font Size: 0 — Prev Excerpt | Next Excerpt — Selection: (34866-34866) Create Excerpt

Selection Info

- mpm1-2009 (34840-34999)
  - meta-level comment abo...
- mpm1-2009 (34864-34868)
  - emotion or value words

Codes

- comment type
  - clarification
  - concept
  - conjecture
  - errors
  - example
  - explanation
  - extension to the problem
  - goals
  - justification


value words
hypothesised cases
language
hypothesised cases
metaphors
reference to a previous comment
hypothesised cases
social pleasantries
simple supporting case
humble language
Profile of example
context
simple supporting examples
conjecture exploration
worst-case scenarios
proof strategies
hypothesised cases
notation, re-representation, meta, explanation, justification, induction, questions
counterexamples
explanation, asking questions, proof development, errors

Applying NLP Machine Learning to Mathematics

What?

- Using pattern-learning and data sets.

Why?

- To handle natural language tasks.
- To participate in mathematical dialogue.
- As a pathway to intuitive approaches, and to creativity.

Where are we today? Early explorers!

Analogy with NLP Development

Consider the development of NLP:

10 years ago - e.g. 2009 rap lyrics generator:

get some to go yeah baby she got it she got it she got it i do my thang in the club you can do it

Today - e.g. GPT-2:

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, “In South America, such incidents seem to be quite common.”

Data: Math StackExchange

982,338 questions

1,379,347 answers

750,000 questions with at least one answer

450,000 questions with an accepted answer

This dataset is available for download:

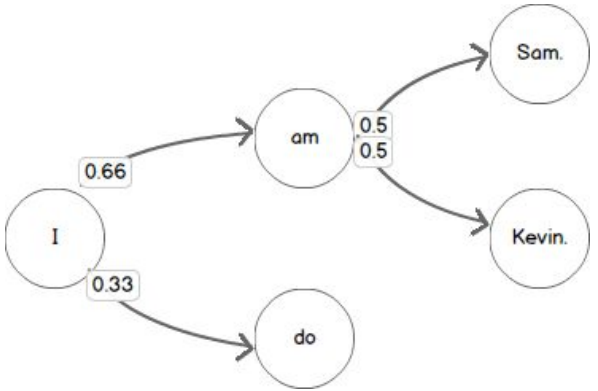
https://zenodo.org/communities/egbot

We also considered: Quora, Reddit, and MathOverflow

Algorithm 1: Markov Model / n-gram

Learns word sequences.

This was the standard NLP technique 5 years ago. Fast to train. Used here as a benchmark for more complex models.



Algorithm 2: Case-based

1. Given an input question Q
2. Look for the most similar past Q', and respond with the matching answer A'.
3. Apply regex modifications to match A' to Q.

Built on ElasticSearch

A search and analytics engine with customisable analyser and similarity settings.

Good framework to use for text mining as it performs fast preprocessing and scoring of the data.

Algorithm 3: Predict-next word neural net

£300

LTSM (long term short memory) deep learning model.

- Recurrent structure for sequences.
- Chooses what features to hold in memory as it reads.
- Can learn intermediate abstractions and rules.
- C.f. Attention-based Transformers (e.g. GPT-2), which have a more flexible memory.

Cons: Slow to train! 1 month for a fairly small network on a consumer-grade GPU. This limited our exploration.

Algorithm 4: General Purpose deep-learning Text model GPT-2 (medium)

A big (345 million weights) deep network.

There is a bigger, better version (but its not available).

Learns P(next-word | previous text)

Attention-based Transformer architecture.

Trained on 8 million web pages covering a wide range of topics.

No specific maths training.

Evaluating a Generative System

Quality?

- Novel / interesting output?
  - Subjective and high time cost
- Correct output?
  - High time cost
- Real-world crowdsourcing? e.g. Google
  - First build a useful system.

Quant?

- Measurable tasks drive progress.
- Accuracy at…?
- ...Recognising the correct answer from a set of sampled answers.

Performance Results: Quant

Baseline (random):	20%
Markov model:	45%
LSTM:	40%
Case-based model:	60%
GPT-2:	-

Performance Results: Quality

Markov model and LSTM: like stopwords, and money. E.g. *“$ $ $”*

Case-based: Great if correct - but it’s pure memorisation.

GPT-2 - Gibberish - but topic, grammar, and language are intriguing

> What is an expected value?

The expected value (EW) is the sum of the logarithms of x multiplied by the sum of the square roots over all values of y.

It is often stated that you should think of the expected value as the sum of your current situation and the probability of that situation changing in the future. The more severe your situation is, the greater the expected value of your situation.

You can use the expected value of your situation to decide how your current situation is going to change. There are some mathematical formulas and methods to predict what your current situation will be like tomorrow:

- *The Bayes rule*

More from GPT-2

> What is the Gamma function?

The Gamma function gives you an estimate of the speed of your computer, and is usually given by the formula \[\begin{align*}\frac{f}}{\cdot \frac{c_k}{\cdot g} = \frac{100}{f}\frac[g - c_k]^{1}\left[\frac{100}{f}\frac{c_k}{\cdot g}\right]} \end{align*}

I believe this is very much like the speed of light.

GPT-2 imagines a philosophy - science conversation

GPT-2 > Give me an example of binomial distributional theorem.

Derrida: This, actually I think is a problem with the argument that "I like it, that's why I like it!" You'd have to actually do this with a really bad sample of people.

Friedman: When you were using an actual number you'd want to sample the population of your sample, and there were quite a lot of people. When you could actually sample it and have a very small sample, it was like: "We have a problem here!"

Derrida: But there are some areas of statistical thinking as you go, like statistics and models, where you can find this particular solution.

Friedman: And it's not just on the data. I am very grateful for a lot of work by Richard Thaler (as well by others that I've read and admired over my long career). He pointed to this

Notes towards a Hybrid System

This project did not produce a useful system. But these explorations help suggest designs for one.

NLP is poor at learning algorithms/procedures…

...Suggesting rather than solving.

...Look at hybrid systems, linking NLP intuition with TP and CAS.

Computing power matters…

...Use transfer learning to leverage expensive deep-learning models.

Summary

The philosophy of mathematical practice provides a starting point for studying mechanisms by which research mathematics progresses -- as messy, fallible, and speculative as this may be.

Empirical work and user studies confirm that examples play an important role in real-world mathematics.

We can usefully study ``backstage mathematics", extracting principles which are sufficiently clear as to allow an algorithmic interpretation.

There are a wide range of methodologies which can be used (and triangulated) in this context.

A focus on aspects of mathematics other than proof seems to be promising (especially for ML approaches).