Natural Language Math Problem Assistance Tool

Stanford Lockhart
Chelsey Childs
Niclas Skaalum
Geoff Caven

Problem Statement

The problem that we are basing the project around is that of taking an utterance of a math problem (e.g. “what is two plus two”, “two x equals 6, what is x”) and transforming it into a symbolic or semantic representation and finally into a single result.
When parsing the utterances, we will have to solve ambiguities, like whether “minus three plus two” means “-3+2” or “-(3+2)” (Liang & Potts, 2015). Once the parsing of the utterances is completed, the problem will be to display these using symbolic notation, and solve the resulting equation.
The final part of problem will deal with variables, and carrying these variables from one query to the next. In this step, we want to be able to solve utterances of the form “two x equals 6, what is x squared plus 1”.
As a final stretch goal for the project, being able to plot functions that contain variables would be a useful tool to have, solving utterances of the form “plot y equals x”.

Approaches

The first approach in solving the problem will be to utilize a grammar to take the utterances and transform them into a prefix semantic representation (Liang & Potts, 2015). Parsing the language into symbols will require knowing the different words that can be used to describe an operation, i.e. “times”, “multiply”, “by” all mean to multiply. The implementation will focus on arithmetic and elementary algebra. Technical jargon terms and mathematic questions requiring calculus or above skills in mathematics will be excluded from the implemented program.

Most of the parsing can be done using the nlptk package for Python (Bird, Klein & Loper, 2009). By using this package, we will be able to do part of speech tagging on the utterances that we are given. One of the tags, CD, denotes numbers - the other tokens can then be parsed to see what operation they refer to, and then build the prefix notation from that.

Using a dictionary of mathematical terms will help to resolve these words into operations (Wells, 2003). Once the utterances are in prefix notation, it will be trivial to solve them, so the majority of the work will be in parsing the utterances and ensuring that any ambiguities are either resolved or given a default result with instructions to the user on how to get another result if desired.

The addition of variables will mean an addition to the grammar used to parse the utterances. Instead of just having variables such as “x”, the problem would benefit from being able to solve word problems. For this, we would require much more advanced parsing to create a solvable system of equations (Kushman, Artzi, Zettlemoyer & Barzilay, 2014).

Project Plan

Initially, we plan to decide on a common language or framework to develop in. We will likely develop in a lightweight scripting language with good regex support, however our decision may change on future investigation.

Once we begin development, our first task will be to interpret a fixed set of natural word fragments to recognize English numbers and math operands. For example, the “teen” fragment may be used to recognize values 13 to 19.

With the ability to recognize numbers and operands, the solution will then require logic to solve utterances expressing simple math problems involving two numbers. When this logic is sufficiently fleshed out, we can proceed to more complex utterances involving multiple expressions and ensure to follow order of operations principles.

Once the solution is capable of solving arbitrarily long math problems expressed in natural language, we plan to polish and refactor the solution to ensure we have a working product by the end of the project timeline. Given additional time, we will proceed to develop stretch goals including variable handling and function plotting. We plan for our solution to function in command line, however if we implement function plotting we will also need to implement a rudimentary visualization tool to display plots to the user.

References

http://www.abstractmath.org/Handbook/handbook.pdf
Wells, C. (2003, March). A handbook of mathematical discourse. PA: Infinity.

http://annualreviews.org/doi/pdf/10.1146/annurev-linguist-030514-125312
Liang, P., & Potts, C. (2015). Bringing machine learning and compositional semantics together. *Annu. Rev. Linguist.*, *1*(1), 355-376.

https://homes.cs.washington.edu/~lsz/papers/kazb-acl14.pdf
Kushman, N., Artzi, Y., Zettlemoyer, L., & Barzilay, R. (2014). Learning to automatically solve algebra word problems. Association for Computational Linguistics.

http://www.foo.be/cours/dess-20122013/b/Natural%20Language%20Processing%20with%20Python%20-%20O'Reilly2009.pdf
Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.