matics and Cybernetics, Baden-Baden, invited paper, 1996. To appear.

Understanding and Solving Algebra Story Problems by Neural Networks and Computer Algebra Systems

Karsten Homann & Anita Lulay
Universität Karlsruhe · Institut für Algorithmen und Kognitive Systeme
Am Fasanengarten 5 · 76131 Karlsruhe · Germany
{homann,lulay}@ira.uka.de

Abstract

The research towards the development of mathematical assistants for high level mathematical problem solving has led to environments integrating specialized mathematical packages by combining techniques from artificial intelligence and symbolic mathematical computations. Although claiming to assist in any kind of mathematical problem, these environments can not understand and solve algebra story problems. Such problems can be tackled by understanding, abstraction, and transformation of their representation into symbolic equational form which can be solved by algebraic algorithms. We report on a hybrid architecture in which the process of understanding, representation, and answer generation is done by neural networks. The integration of the learning capabilities of neural networks into natural language processing systems combined with the symbolic algorithms of computer algebra systems provide a powerful tool for solving algebra story problems.

Keywords

Problem Solving Environments; Neural Networks; Symbolic Computation

Introduction

Understanding and solving algebra story problems is of interest for various applications in cognitive psychology, artificial intelligence, mathematics education and cybernetics. Among others, important questions consider theories of problem solving strategies, language comprehension, recognition of semantic relationships, children's problem solving capabilities, memory organization, linguistic factors.

Algebra story problems are an important practical application of mathematics since real-world problems usually do not arise in terms of equations but as verbal or pictorial representations. The problems are solved by understanding, abstraction, and transformation of these representations into symbolic equational forms which can be solved by algebraic algorithms. The transformation formula $\mapsto$ result is task of computer algebra systems (CAS) which are increasingly used in teaching and performing mathematics. Therefore, the transformation problem $\mapsto$ formula becomes more significant.

DISCERN (Miikkulainen, 1991) has shown remarkable success in understanding narratives about stereotypical event sequences built entirely from distributed artificial neural networks. The learning capabilities of neural networks offer new possibilities not provided by the symbolic approaches, e.g. extension of the lexicon, extraction of word semantics, memory organization and problem-solving templates from examples.

Several studies have classified word problems in algebra or physics by the form and complexity of the underlying mathematical equation and story (Kintsch and Greeno, 1985; Mayer, 1981). These stories have corresponding templates consisting of variables, operators, values, and relations of a problem. Figure 1 gives a schematic overview of an architecture for understanding and solving story problems with characters, actions, and objects, e.g. *The Bounty travels 5 hours downstream. The current is 6 km/h. The Bounty returns against the current in 10 hours. What is the speed of the Bounty in still water?*

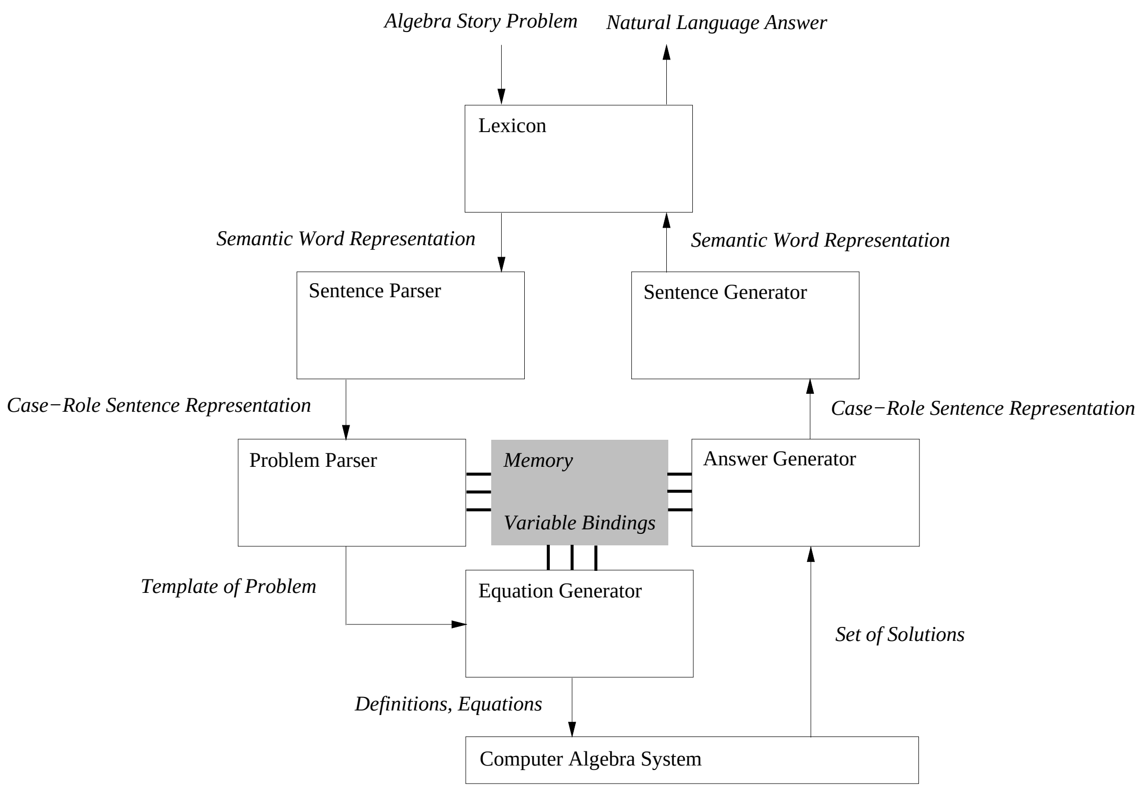

Figure 1: Schematic overview of the architecture

The following sections briefly describe the structure analysis of algebra story problems, the word representation in a connectionist lexicon, parsing of sentences and problems, and how problems are solved by generating equations and passing them to a computer algebra system.

Algebra Story Problems

Usually, one can find five formats of problems in algebra textbooks: equation, formula, number, arithmetic word, and story. While arithmetic word problems only consist of a story line involving addition and subtraction, story problems additionally include characters, objects, and more complex actions based on sophisticated source formulae involving rate, geometry, physics or statistics.

A first step in solving story problems is determining the category of the underlying mathematical problem, e.g. ratio, angle, motion, current, work. The algebra story problems of textbooks used in secondary school were selected and classified into families, categories, and templates by (Mayer, 1981). Our approach aims at solving algebra story problems corresponding to the introduced templates.

A template is a description of a propositional structure of a story problem. Propositions consist of variables (time to travel downstream), operators (twice), values (number), and relations (equal). Two story problems are of the same class if they have a common template, i.e. they share the same list of propositions, independently of the values of the variables.

An appropriate template for the introduced example is given in (Mayer, 1981):

Current: Round trip 1
Time with current = 5 hours
Time against current = 10 hours
Rate of current = 6km/h
Rate in still water = FIND
Modifications: story line with boats and airplanes

The main drawbacks of this approach are:

- The variable with an unknown value is fixed within each template (e.g. a new template is necessary to determine the rate of current).
- Any possible combination of given variables requires a new template (e.g. distance given instead of time).
- It is difficult to determine the template to a given problem (e.g. impossible without considering the whole story).

These points were not important for a theoretical categorization of story problems but are necessary for automated classification and solving. They can be avoided by extending specialized templates to a more general one. This new template corresponds to a more general class of story problems, all based on the same set of source formulae. It includes all possible variables which may occur in a class of problems. The variables with unknown values are no longer fixed. The more general template can be determined easier (e.g. by keyword current).

Lexicon

We adopt the word representation in a global lexicon using FGREP (Forming Global Representations with Extended Backpropagation) introduced in (Miikkulainen and Dyer, 1991). The FGREP architecture consists of a three-layered backpropagation network and a global lexicon which stores the learned representation of words. The representation is not based on predefined feature vectors and is context dependent. A new context implies a change in the representation.

The lexicon consists of two 2-dimensional self-organizing feature maps (Kohonen, 1988) connected through associative links. These links learn a mapping from the lexical map to the semantic map and vice versa storing the orthographical and semantic representations of the words of the algebra story problems. Orthographically similar words are mapped onto neighboring nodes of the lexical map (e.g. `BOAT, ROAD`) while words belonging to the same concept (e.g. `boat, plane` to `carrier`) are represented near each other in the semantic map (figure 2).

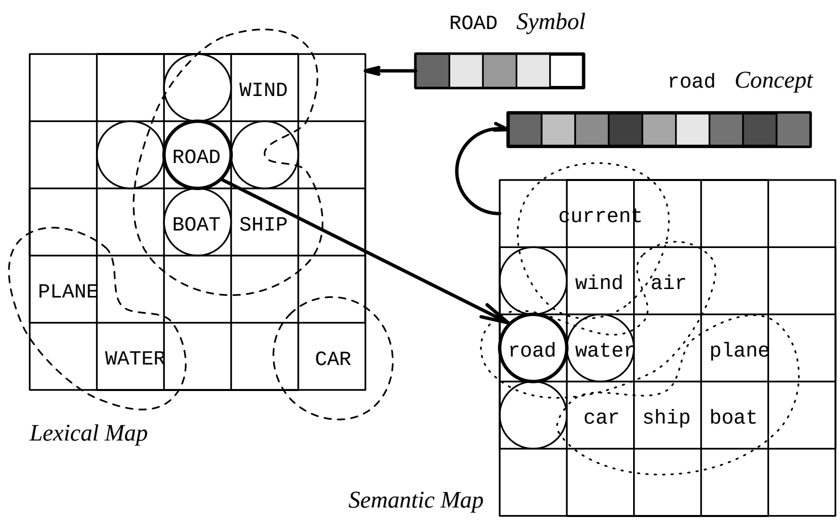

Figure 2: Architecture of the Lexicon

The problem of distinguishing between instances of the same concept can be avoided by the ID+content technique. However this technique cannot be used for the representation of numbers.

Parsing Algebra Story Problems

The sequence of concept representations determined by the lexicon is passed to the parser which produces an internal representation of the story problem. The parser is implemented as a recurrent network with two hidden layers. The activity is propagated to gradually form the output by recurrent FGREP. The parser consists of two modules (Miikkulainen, 1991): a sentence parser which determines the case-role representation of an input sentence and a problem parser which computes the template representation of the whole story.

The case-role representation of a sentence consists of the following roles: agent, action, quantity, value, location, loc-attribute; e.g.

The Bounty returns against the current in 10 hours.

Bounty	returns	time	10	current	against
Agent	*Action*	*Quantity*	*Value*	*Location*	*Loc-Attr.*

When the appropriate template of the story problem is found, it is filled by the problem parser, which has the same structure as the sentence parser. It receives as input a sequence of the case-role representation of a sentence and fills the template of the whole problem at its output layer. The template for the introduced example consists of rate of current (v_c), distance with current (d_w), distance against current (d_a), total time (t_t), rate in still water (v_0), time with current (t_w), time against current (t_a), rate with current (v_w), rate against current (v_a), time difference (Δt), and rate difference (Δv), e.g.

6				?	5	10				
v_c	d_w	d_a	t_t	v_0	t_w	t_a	v_w	v_a	Δt	Δv

Filling the template is the last step in understanding a given algebra story problem.

Solving Algebra Story Problems

Solving algebra story problems requires three steps: (1) determination of the underlying mathematical equations of a template, (2) calculation of the solution of a given equation by a CAS, (3) generation of the natural language answer.

There are several possibilities to select equations. Since many fillers in the template representation are empty it is possible to determine the special type of the problem by a simple backpropagation network. This allows to select only a small subset of the set of equations of a template. Another possibility is to pass the whole set of equations to the CAS without further selection. This strategy has several advantages:

- It is not necessary (and very expensive) to store all subsets of equations for every specialized template of a problem category.
- Incomplete input problems can be solved by a CAS symbolically (e.g. $v_0 = 2 * v_c$).
- It avoids the difficulty to find the specialized template of problems containing redundant information. Contradicting information can be detected by the CAS when passing the whole set of equations.

The set of equations of the templates are stored in the memory together with the corresponding variable bindings, case-role representation of questions, and the template representation of the problem.

CAS's like MAPLE, MATHEMATICA offer algorithms for symbolically solving sets of equations, e.g. $Solve(\{v_w = v_0 + v_c, v_a = v_0 - v_c, v_w = \frac{d_w}{t_w}, v_a = \frac{d_a}{t_a}, t_t = t_w + t_a, \Delta t = t_a - t_w, \Delta v = v_w - v_a, v_c = 6, t_w = 5, t_a = 10\}, \{v_0\}) = \{(v_0 = 18)\}$.

The translation of the solution into natural language will be done by an answer generator using the memory to produce the case-role representation of the answer which is created by the sentence generator and lexicon.

Conclusion

A hybrid architecture for understanding and solving algebra story problems is given. The process of understanding, representation and answer generation is done by neural networks. Some of the interesting aspects of this approach are: modeling of higher-order cognitive tasks by neural networks, algebra story problem solving as an application of NLP, translation and representation of story problems in sets of equations by neural networks, design of representations of arithmetic operations and numbers in connectionist NLP.

Further efforts must be investigated to enhance the applicability and combination in problem solving environments, the equation generator and the interaction with a CAS, to allow more complex input sentences, extension of the connectionist lexicon to include more linguistic information, e.g. about lexical categories, embedded usually in a symbolic lexicon, influences of modifying the lateral connections between neurons of self-organizing feature maps on the representation of ambiguous words, new techniques to distinguish between instances of the same concept.

The integration of the learning capabilities of neural networks into NLP systems combined with the symbolic algorithms of CAS provide a powerful tool for solving algebra story problems.

References

Kintsch, W., Greeno, J.G. (1985); *Understanding and Solving Word Arithmetic Problems*; Psychological Review, 92(1) (pp. 109–129)

Kohonen, T. (1988); *Self-Organization and Associative Memory*; Springer-Verlag

Mayer, R.E. (1981); *Frequency Norms and Structural Analysis of Algebra Story Problems into Families, Categories, and Templates*; Instructional Science, 10 (pp. 135–175)

Miikkulainen, R.P. (1991); DISCERN*: A Distributed Artificial Neural Network Model of Script Processing and Memory*; PhD thesis; UCLA

Miikkulainen, R.P., Dyer, M.G. (1991); *Natural Language Processing with Modular PDP Networks and Distributed Lexicon*; Cognitive Science, 15(3) (pp. 343–399)