A Question Answer System for Math Word Problems

Christian Liguda and Thies Pfeiffer

Artificial Intelligence Group, Faculty of Technology, Bielefeld University
{cliguda, tpfeiffe} @ techfak.uni-bielefeld.de

Zusammenfassung Solving word problems is an important part in school education in primary as well as in high school. Although, the equations that are given by a word problem could be solved by most computer algebra programs without problems, there are just few systems that are able to solve word problems. In this paper we present the ongoing work on a system, that is able to solve word problems from german primary school math books.

1 Introduction

"A primary school has won 30 table tennis rackets at a competition. Two more are donated by a teacher. To each racket belong 3 balls. Everything should be shared among 8 classes. How many table tennis rackets and balls do each class receive?" (translated from [Hans Bergmann, 2010], question added). According to Franke and Ruwisch word problems like this describe situations in which mathematical relationships are embedded within a short story ([Franke and Ruwisch, 2010]). Therefore word problems are different from pure arithmetics, because the calculation that has to be carried out must be extracted from the given text. The motivation for such problems is to teach children how to explore their environment with mathematical tools. Also the equations that are given by a word problem could be solved by most computer algebra programs without problems, there are just few systems that are able to solve word problems like the one above (see Section 3).

In this paper we present the ongoing work on a program which is able to solve some kinds of word problems from primary school math books. The motivation is to understand which steps are necessary for a program to solve mathematical problems that are given within a short story and how they could be solved. Furthermore we hope to get insights that could be transferred to similar domains, like word problems of high school math, physic problems or logic problems that are presented in natural language and embedded in a short story.

At first we analyze the solving of word problems in more detail in Section 2 to find out, which tasks have to be mastered in order to solve such problems. In Section 3 we discuss former systems that were able to solve word problems and how they handled the identified problems. Afterwards we take a closer look at an approach that is found in most modern systems and show the difficulties of this approach. An overview of our system is then given in Section 5. In the next two sections our model for representing word problems and our approach for transforming a given text to this model are presented. In this sections we also show how our approach could deal with the identified problems that are discussed in Section 2 and Section 4 before. The paper ends with a discussion of our system in Section 8.

2 Problem Analysis

In order to build a system that is able to solve word problems from current math school books, it is important to understand which tasks must be mastered, by such a system. [1]

(P1) Wide range of surface characteristics: Often children (and computer programs) just look at surface characteristics of a given text to solve word problems, without understanding to story of the text or the relations of the given mathematical concepts [Stebler, 1999]. Together with some quite simple heuristics they try to solve word problems. These heuristics could be rules like *"if a text contains the word "left" the numbers in the text should be subtracted"* or *"a big and a small number implies to subtract the smaller one from the bigger one."*. To avoid this behaviour modern school books contain word problems with a wide range of surface characteristics.

(P2) Flexible sentence structure: A motivation for teaching word problems is to teach children, how everyday situations are related to mathematical relations. Therefor nearly every word problem in modern math books (like the already mentioned [Hans Bergmann, 2010]) describes a different short story and the stories are mostly build up of different sentences. Therefore no fixed sentence structure could be assumed. Furthermore the relevant informations are not always given in the some order, like for example the first set represents a given set and the next set the one that should be added or subtracted from the first.

(P3) Irrelevant information: In addition to the information relevant to solve the word problem, some problems contain irrelevant information. Studies have shown that children often have problems with these extraneous information (e.g. [Cook, 2006]).

(P4) Different concepts: Strongly related to the last point is the appearance of different concepts in a word problem. In word problems at primary school level the concepts *set*, *money*, *time*, *weight*, *distance* and *volume* are found. Furthermore in some word problems numbers are embedded, which could not be part of a calculation, like the number in the expression *Class 3a*.

[1] The following issues describe just some aspects of the problem solving task. For a more detailed analysis take a look at [Franke and Ruwisch, 2010] or [Verschaffel et al., 2010]

(P5) Multistep problems: Word problems could be distinguish at whether they could be solved with one arithmetic operation or if more than one is needed [Reed, 1999]. Problems that need more than one operation to be solved, require planning and the stepwise solving of subgoals, until the problem is solved completely.

(P6) Understanding the meaning of words: For some problems it is necessary to understand the meaning of a word. For example a passenger could be referenced as person later, or it is necessary to understand that boys and girls are children.

3 Related Work

The first system, that was able to solve math word problems, was the program STUDENT [Bobrow, 1964]. The main idea for understanding natural language was the use of sentence templates of the form *?x* `plus` *?y* or `the product of` *?x* `and` *?y* to identify mathematical concepts. Together with some preprocessing and splitting of complex sentences, a word problem like "*If the number of customer Tom gets is twice the square of 20% of the number of advertisement he runs, and the number of advertisements he runs is 45, what is the number of customer Tom gets?*" was transformed to prefix notation and solved by some techniques of solving simultaneous equations. STUDENT was able to solve some word problems from high school algebra books, but was limited to those, that use special sentence structures and keywords (Problem (P1) and (P2)). STUDENT did not really needed strategies to deal with different concepts or strategies, because STUDENT took whole parts of a sentence as name for a variable and was not confused by some irrelevant information, because they were hold in variables that were not referenced later. In doing so, STUDENT was able to handle irrelevant information, but could not detect completely different words as synonyms, or hierarchical relationships among words (P6)

The System WORDPRO [Fletcher, 1985] was based on a psychological model, that was developed by Kintsch and Greeno [Kintsch and Greeno, 1985]. The main idea was to organize relevant information about sets in frames with the slots `Object`, `Quantity`, `Specification` and `Role`. In the slot `Specification` information about owner, time or location of the set could be stored. The slot `Role` could contain roles like `superset` oder `transIn`. Afterwards the problem solving was done by a rule based system. Each rule (also named as schema) checked, if some preconditions in the slots of the frames were given, and when they were fulfilled, a rule could do a calculation and added the result to a slot. In doing so WORDPRO could solve some problems of addition and subtraction, but had no parser and needed propositions as input. By including information about location, time and owner WORDPRO could deal with some irrelevant informations (P3), but had difficulties with all other mentioned problems.

The approach of organizing informations in frames and solving the problem with schemata is the one that is found in nearly every later system . The latest is MSWPAS ([Yuhui et al., 2010]) which is able to solve multi-step word problems containing addition and subtraction. Mukherjee and Garain give a more detailed review about the mentioned systems (except MSWPAS) and others [Mukherjee and Garain, 2008].

4 Discussion of the Frame and Schema Concept

Like mentioned in Section 3, most modern systems are based on the frame and schema concept. Although they are able to solve addition and subtraction multistep word problems, there are the following problems with this concept:[2]

(FP1) **Relations:** A frame combines several information about a set in a single object. Relations to other sets could be specified using the slot `Role` with attributes like `subset` or `transIn`. The problem is that these attributes are too general, because the attributes do not specify for what other sets the relation holds. This is important, if irrelevant information are given, or the problem contains several sets that are in different relations to each other (see. P3 and P4).

(FP2) **Multistep Problems:** Generally it is possible to solve multistep word problems with schema how MSWPAS demonstrates. If the chronology of the arithmetic operations is different from the order in which the rules are activated, intermediate data has to be stored. Systems that are based on the frame and schema concepts, do not give a solution for this problem (see P5).

(FP3) **Deadlock Problem:** A schema does not consider the relation to other schemata, but has just a local view on the frames. If a system would try to solve the problem *"Bob gives 7 marbles to Tim. Afterwards Bob has half the marbles he had at the beginning. How much marbles had Bob at the beginning?"* no schema could be activated. Because the number of marbles Bob has at the beginning as well as at the end are unknown, neither the `transOut` schema nor an assumed `factor` schema could be activated.

5 Overview

In order to solve the problems mentioned above our system uses some different approaches:

1. **Information Extraction:** The systems starts with stepwise annotating the text. At the end of this process all relevant information are annotated. In most cases a small selection of annotations will be enough for building.
2. **Building an Augmented Semantic Network:** On the basis of the annotations an augmented semantic network is build. The nodes represent mathematical concepts and the edges relations between them.
3. **Transformation of the Model to Equations:** Each edge represents a relation that could be transformed to one or more equations.

[2] The abbreviation FP stands for Frame Problem

4. **Solving the Equations:** The generated equations are solved by an external computer algebra program.
5. **Modify the Model if necessary:** Sometime the model contains to little or to much information to assign a value to a variable. Therefore our system is able to update the model in small steps to correct some common mistakes from the building process.

In the next section we present the concept of the Augmented Semantic Network as well as the transformation to a system of equations.

6 The Formal Representation of a Problem

An Augmented Semantic Networks for representing for word problems is primary motivated by (FP1). By using a semantic network we make relations to other concepts explicit ([Simmons, 1973]).

6.1 Node

Nodes represent measurable quantities, which are the already mentioned concepts. We use different kinds of nodes to represent them. The attributes of a node depend on the type of note. A Node of the type set has the attributes `Object`, `Quant`, `Location` und `Owner`.

6.2 Situation Counter

Every attribute, that represents a measurable quantity, has as domain a list of tuple. Each tuple consists of an index and a value. In doing so a node could represent a attribute at different situations by using different indices for the attribute. Our system uses a situation counter per node, that is incremented when a relation changes the value of an attribute. With this new approach we can better represent the changing over time and are able to solve the problem (FP2).

6.3 Edge

In contrast to normal semantic networks where edges are just connections between two nodes in our system they are augmented with some additional information. Like mentioned above a node can have more attributes that represent measurable quantities in different situation. Therefore a edge must contain informations, to what attributes and situation it belongs. Furthermore some edges represent relations between concepts in which some additional informations are important. The relation `factor` is an example for this type of relation. An additional parameter containing the number 4 is needed to represent the relationship of the text *"For each window 4 bolts are needed."*.

6.4 Examples

In Figure 1 the semantic network of the word problem *"A primary school has won 30 table tennis rackets at a competition. Two more are donated by a teacher. To each racket belong 3 balls. Everything should be shared among 8 classes. How many table tennis rackets and balls do each class receive?"* is given.

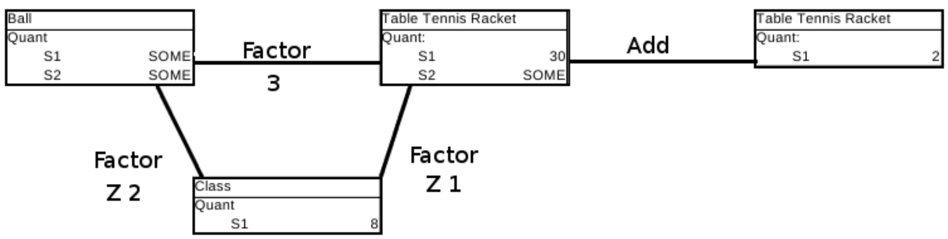

Abbildung 1. Augmented Semantic Network of a word problem

The variables Z_1 and Z_2 represent the unknown factors of the relations. Unknown values for the attribute `Quant` in a situation are labelled as `SOME`.

6.5 Transferring the Network Model to a Set of Equations

Because the approach of solving a model with a rule based system leads to several problems, our system uses a different approach. Like mentioned in Section 5, every relation is transformed in a set of equations. More than one equation is generated when an edge is connected to more than one attribute in one situation. Like mentioned in section 6.2 this new approach allows to solve problems in which the chronology of the arithmetic operations is different from the order in which the values of a variable at different situations can be calculated. Therefore this strategy enables the system to solve (FP 2). In the example of section 6.4 the attribute `Quant` of the set *Table tennis racket* contains two situations. Therefore the `factor` relation is transformed to the two equations:

$$Y_0 = 3 * 30$$
$$Y_1 = 3 * X_1$$

The variables Y_0 and Y_1 represent the two values for the attribute `Quant` of the set *balls*. X_1 is a variable for the unknown value of the attribute `Quant` of the set *Table tennis racket* after the addition. A central variable management unit assures globally unique names for all variables.

The generated set of equations can be solved with standard techniques of computer algebra [Geddes et al., 1995]. Therefore we use the open source computer algebra program Sage [Stein et al., 2011] to solve the equations. Using an external program has the following advantages:

- Sage is able to solve complex equation systems. Through this our system is able to solve multistep word problems (P5 and FP2).
- No special work has to be done to solve equation systems, where dependencies between equations have to be taken into account (FP 3).
- In case of further improvements of our system no special work has to be done for solving the generated equations.

Because of the explicit relations between concepts and the differentiation into different attributes and situations irrelevant informations are no problem for the solving process, if they are correctly represented in the model.

6.6 Update the Model

Sometimes the building process, which is described below, is ambiguous. One consequence is that the model contains to little or to much information. Sometimes this makes a model unsolvable. For example problems in which a set has no inital value can be interpreted in different ways. Sometimes it is assumed that the inital value is zero and sometimes the value could be calculated by some constrains. If the model could not be solved our system is able to redefine the model. The redefinition is done stepwise and changes, that did not change the solvability, can be revoked.

7 Natural Language Processing

As per description in Section 2 pure keyword matching has the disadvantage to be very shallow. Predefined sentence templates on the other hand are to strict. As compromise we use information extraction techniques (vgl. [Jurafsky and Martin, 2009]) to get the relevant informations which are afterwards transformed to a model.

7.1 Information Extraction

Information Extraction defines the task of transforming unstructured informations, that appear within a text, into structured data [Jurafsky and Martin, 2009]. Without understanding the whole sentence our system tries to find informations about the different concepts and relations between them.

By using the open source toolkit Gate [Cunningham et al., 2011] we created a pipeline for language processing. In this pipeline the document is stepwise annotated. The ability to access all annotations of all previous steps allows the creation of very complex annotations by referring to earlier ones. The Pipeline consists of the following components:

- **Document Reset:** Reconstruct the original state of a document.
- **Sentence Splitter:** Add an annotation at the end of each sentence.
- **Tree Tagger:** Wrapper for the TreeTagger ([Schmid, 1994] and [Schmid, 1995]) to identify the part of speech for each word. In addition the lemma for every word is identified.
- **RF Tagger:** Wrapper for the RFTagger ([Schmid and Laws, 2008]), to specify gender and number of nouns.
- **Case Tagger:** Wrapper for the CaseTagger ([Perera and Witte, 2005]) to identify the case of a noun.
- **NumberTagger:** A Gate plugin for annotating numbers. Even numbers in written form are annotated.
- **Identify Persons and Concepts:** Identifies persons and furthermore words that are associated with a concept by using wordlists (e.g. the word *meter* that is associated to the concept *distance*).
- **Identify Modifier:** By using wordlists the system identifies words which can be used to define the role of a set. For example the word *together* implies that the set in the sentence is a superset.
- **Modifier Cleanup:** Some wrong modifiers are removed.
- **Identify Lemma:** The lemma of every verb is matched against wordlists to identify relevant relations. For example the lemma *buy* implies a subtraction of money.
- **General Identifier:** For this component we use the JAPE language, which is part of a Gate plugin [Cunningham et al., 2002]. In JAPE it is possible to define regular expressions over annotations and use the founded matches for further annotations. At the moment 56 rules are defined to identify personal pronoun, sets, currencies, relations and the role of a set.

7.2 Building the Model

After the original text has been annotated step-by-step, in most cases the information from a selection of annotations are already sufficient to create the semantic network model to represent the hidden math problem. The current implementation does only support sets, so the three annotations Person, Relation and Set are adequate. The annotations are interpreted in order of appearance in the text. When a person annotation is interpreted, the name of the person is added to a list of persons relevant for the problem. This is necessary if, e.g., a set has to be distributed along the people mentioned in the text.

When a set annotation is interpreted, the set is added to the model and the role of the set is interpreted in the context of the model. In this step, one or more relations are drawn to other sets already available in the model. The model also differentiates between sets and super-sets. If applicable, a relation `contains` of the super-set to the sub-set is established. This is necessary, as some problems define multiple sets which are all contained in a super-set, which itself has a `factor` relationship to yet another set.

An annotation of a relation may refer to more than two sets, depending on the number of sets in a sentence. The interpretation of an annotation of a relation is done in 3 steps:

– Search in the model for all sets, that are relevant for the annotation. If a set does not exist it must be created.
– Identify the attributes and situations for the sets.
– Add a relation of the corresponding type for each pair of sets for each attribute and situation.
– Sometimes a role for the created sets is also available. If so, the role of the sets is interpreted as if the set was added to the model.

8 Discussion

The presented system for solving math word problems goes beyond previous versions. As one essential part it employs a natural-language processing pipeline that augments the input text with annotations. This is done in a step-by-step manner until in the end all relevant information have been extracted, without relying on stereotypes and patterns and without a deeper understanding of the semantic structure of the text. Another essential difference is the transition from a frame-based approach to augmented semantic networks. This way, the relationships between individual concepts can be represented more precisely, which also allows a more robust detection of irrelevant information. The augmented semantic network is also extended to support the representation of the temporal order of actions by implementing a state system for attributes and values of nodes. The scalability of this approach to differnt types of math problems is ensured by a defined mapping from the semantic network to a system of equations which is then solved by an external computer-algebra program. This allows math problems of higher complexity to be solved than what is representable without problems in schemes.

There are still a lot of questions unaddressed. First of all, the concepts of money, distances, time and volumes are not handled in the current implementation. The framework, however, is already designed to support these concepts. To approach P6, the problem of understanding the meaning of words, the GermaNet [Hamp and Feldweg, 1997] or Wiktionary ([Zesch et al., 2008]) could be accessed, to identify the most common problems (see description of P6). For the evaluation of the system it is planned to compile a corpus of math text problems.

An interesting follow-up would be a system for generating questions and answers related to the math text problem in natural language. The current version is already able to detect unknown values in the model, but it cannot generate natural language sentences to describe them. The most critical part of the system is the collection of rules in the information extraction system. While the current system is already able to solve many individual problems, it has not been evaluated yet and we expect that several more rules need to be added while doing so. In the long run, it seems attractive to move on to more recent approaches for information extraction by training the rules from the created corpus.

Generating a system which extracts information from natural language texts and builds a formal model that is then solved provides valuable insights into the basic underlying processes, which might also be cognitively relevant. Beyond that, it would be interesting to combine this system with a tutoring system, so that the tutoring system can solve and explain math text problems previously unknown to the system. It could also be interesting to use the system to classify given math text problems, for example to rate the complexity of the math problem, the complexity of the problem description or the difficulty of the transfer from the description to the model. This could help teachers in the design of math text problems.

Literatur

[Bobrow, 1964] Bobrow, D. G. (1964). A question-answering system for high school algebra word problems. In *Proceedings of the October 27-29, 1964, fall joint computer conference, part I*, AFIPS '64 (Fall, part I), pages 591–614, New York, NY, USA. ACM.

[Cook, 2006] Cook, J. L. (2006). College students and algebra story problems: Strategies for identifying relevant information. *Reading Psychology*, 27:95 – 125.

[Cunningham et al., 2002] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

[Cunningham et al., 2011] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*.

[Fletcher, 1985] Fletcher, C. (1985). Understanding and solving arithmetic word problems: A computer simulation. *Behavior Research Methods*, 17:565–571. 10.3758/BF03207654.

[Franke and Ruwisch, 2010] Franke, M. and Ruwisch, S. (2010). *Didaktik des Sachrechnens in der Grundschule*. Spektrum Akad. Verl., Heidelberg.

[Geddes et al., 1995] Geddes, K. O., Czapor, S. R., and Labahn, G. (1995). *Algorithms for computer algebra*. Kluwer, Boston.

[Hamp and Feldweg, 1997] Hamp, B. and Feldweg, H. (1997). Germanet - a lexical-semantic net for german. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

[Hans Bergmann, 2010] Hans Bergmann, Elena Hauer, et. al. (2010). *200 Textaufgaben wie in der Schule*. Klett.

[Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and language processing*. Pearson Prentice Hall, Upper Saddle River, NJ.

[Kintsch and Greeno, 1985] Kintsch, W. and Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review*, 92(1):109 – 129.

[Mukherjee and Garain, 2008] Mukherjee, A. and Garain, U. (2008). A review of methods for automatic understanding of natural language mathematical problems. *Artif. Intell. Rev.*, 29:93–122.

[Perera and Witte, 2005] Perera, P. and Witte, R. (2005). A Self-Learning Context-Aware Lemmatizer for German. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pages 636–643, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

[Reed, 1999] Reed, S. K. (1999). *Word problems*. Erlbaum, Mahwah, NJ.

[Schmid, 1994] Schmid, H. (1994). Probabilistic Part-of-Speech Tagging sing Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*.

[Schmid, 1995] Schmid, H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*.

[Schmid and Laws, 2008] Schmid, H. and Laws, F. (2008). Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging. In *COLING*.

[Simmons, 1973] Simmons, R. F. (1973). *Semantic networks: Their computation and use for understanding English sentences*. Freeman, San Francisco.

[Stebler, 1999] Stebler, R. (1999). *Eigenstaendiges Problemloesen. Zum Umgang mit Schwierigkeiten beim individuellen and paarweisen Loesen mathematischer Problemgeschichten - theoretische Analyse und empirische Erkundigungen.* Lang, Frankfurt am Main.

[Stein et al., 2011] Stein, W. A. et al. (2011). *Sage Mathematics Software (Version 4.6.2)*. The Sage Development Team. `http://www.sagemath.org`.

[Verschaffel et al., 2010] Verschaffel, L., Van Dooren, W., Greer, B., and Mukhopadhyay, S. (2010). Reconceptualising word problems as exercises inÂ mathematical modelling. *Journal fuer Mathematik-Didaktik*, 31:9–29. 10.1007/s13138-010-0007-x.

[Yuhui et al., 2010] Yuhui, M., Ying, Z., Guangzuo, C., Yun, R., and Ronghuai, H. (2010). Frame-based calculus of solving arithmetic multi-step addition and subtraction word problems. *International Workshop on Education Technology and Computer Science*, 2:476–479.

[Zesch et al., 2008] Zesch, T., Mueller, C., and Gurevych, I. (2008). Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of LREC*.