Learning to Solve Arithmetic Word Problems Using Sentence Simplification

Vishal Rajpal
Northeastern University
rajpal.vi@husky.neu.edu

Abstract

In order to respond to an arithmetic word problem correctly one needs to understand the question to an extent which allows determining the constraints. These are mostly semantic and are imposed by the question on its answer. Constraints from individual sentences suggest a mathematical operation and when the operators from these sentences are used collectively the answer can be derived. To extract the constraints efficiently, a concept of Syntactic Pattern is introduced which is generated by parsing the sentence using a dependency parser. It encapsulates all the relevant information in a sentence including the subject, verb and object with its quantity. Another important method for this thesis which relies on syntactic patterns is Sentence Simplification. The idea is to have a subject, verb, an object and other necessary parts of speech in the sentence so that it suggests a single operation. Based on the identified patterns a sentence may be simplified to multiple sentences. This would make the classification process easier since the sentences are less complex. To this end, it would be the classifier's job to classify the sentence to a mathematical operator. The identified operators for individual sentences are used to build a mathematical equation for the entire word problem. The results based on 3 datasets are reported and seem promising as compared to the existing systems.

1 Introduction

Answering arithmetic word problems has gained a lot of interest in recent years. The problem is attractive to NLP since the text is concise and relatively straightforward with identifiable semantic constraints. As these problems are directed towards elementary school students, they begin by describing a partial world state, followed by simple quantified updates or elaborations and end with a quantitative question. This information can be mapped

to basic operations(addition, subtraction, multiplication and division) and an equation can be created corresponding to the problem text.

There have been a number of attempts to solve arithmetic word problems through Machine Learning. All the approaches which are not template based (Hosseini et al., 2014), (Roy et al., 2015) and (Roy and Roth, 2015) use different methods to extract similar information. Based on different ways the information is represented, an equation is built for the problem text. The template based method of (Kushman et al., 2014) implicitly assumes that the solution will be generated from a set of predefined equation templates. Some of these methods only solve addition and subtraction problems (Hosseini et al., 2014), (Roy et al., 2015) while (Roy and Roth, 2015) and (Kushman et al., 2014) can solve problems for all operations.

The approach presented in this thesis can solve a general class of addition and subtraction arithmetic word problems without any predefined equation templates. In particular, it can handle an arithmetic problem as shown in Table 1.

Example 1:
For Halloween Debby and her sister combined the candy they received. Debby had 32 pieces of candy while her sister had 42. If they ate 35 pieces the first night, how many pieces do they have left?

Table 1: Example Arithmetic Word Problem.

To derive the solution to this problem, the approach needs to understand that *they* refers to *Debby and her sister*. Hence, the number of candies need to be summed up and then *35 candies* need to be subtracted from the total number of candies.

While a solution to these problems requires extracting information and composing numeric expressions, if the sentence is too complex it is hard to extract information accurately.

At the heart of the technical approach, the novel notion of *Sentence Simplification* is involved. Once the sentences in the problem text are simplified, extracting information becomes easier. Each sentence in the problem is simplified to a level where it consists information which is able to be mapped to a single operator. This allows us to decompose the entire problem to a collection of simplified sentences as operator prediction problems. Each sentence represents the quantitative information with the mapped operator. These predictions(operators with the quantitative information) can be

combined together to form a mathematical equation.

The approach focuses on addition and subtraction problems currently, but learning to classify operators will allow us to generalize the approach to multiplication and division problems as well. In particular, the system was able to solve Example 1 although it had never seen the problem before and required both addition and subtraction operations.

The approach is evaluated on 3 datasets, achieving competitive performance on all of them. The next section describes the related work in the area of automated arithmetic word problem solving. The theory of sentence simplification is then presented, later the information decomposition strategy that is based on it is discussed. Section 4 presents the overall computational approach, including the way classifier learns to classify simplified sentences to operators. Finally, experimental study is discussed followed with a conclusion.

2 Related Work

Most of the previous work in automated arithmetic problem solvers has focused on a restricted subset of problems.The approach described in (Roy et al., 2015) handles problems with all basic operations but makes assumptions about the number of quantities in the problem text and the number of steps required to solve the problem. In contrast, our approach does not make any assumptions about the data in the problem text. Kushman's approach (Kushman et al., 2014) tries to map numbers from the problem text to predefined equation templates and implicitly assume that similar equation forms have been seen in the training data. In contrast, our system does not rely on pre-defined templates and hence is able to solve questions of type which have never been seen before. The approach described in (Hosseini et al., 2014) might be the most related to ours. It handles addition and subtraction problems and tries to predict an operator for the verb in the problem text. Hence, it requires additional annotated data for verb categories. Our approach uses the information of the verb present in the sentence and handles addition and subtraction problems for now, but there is no requirement of additional annotated data.

Most of the methods mentioned above are able to solve problems with all basic operations, but our approach is easily generalizable to all the operators and also it performs competitevely as compared to all other approaches.

3 Arithmetic Problem Representation

Our approach addresses word problems that include addition and subtraction problems. Given a problem text, multiple fragments are extracted from it where each fragment is a simplified version of the information presented in the sentence. We refer to these fragments as simplified sentences. Each fragment is represented based on the Parts of Speech it contains. Below are the parts of speech we consider in our representation:

Adjective: A problem text might have multiple types of same entities. Consider the below example:

A pet store had 13 siamese cats and 5 house cats.
siamese cats
house cats.

Table 2: Entities with Adjective.

Though the example sentence had cats as the Noun, but based on the adjectives*(siamese and house)* there were siamese cats and house cats. We consider them as different Nouns unless the Noun is mentioned without any adjectives.

Cardinal: Cardinals are the numbers present in the problem text. We associate them to Nouns based on the index at which they occur. Mostly, they occur before the Noun and hence associating it is not difficult. Sometimes, the cardinal is a reference to an already occured Noun. Consider the example below:

He bought 2 games from a friend and bought 5 more at a garage sale.
2 games
5 games

Table 3: Entity with Cardinal Numbers.

5 more in the above example refers to games. We associate this to the last quantified Noun encountered in the sentence.

Conjunction: Conjunctions are stored mainly to simplify the sentence having them. Most importantly, the index at which it occurs in the sentence plays a crucial role in creating simplified sentences.

Determiner: Determiner by definition is *a modifying word that determines the kind of reference a noun or a noun group has.*

There are 28 students and Every student has their own lunchbox.
Every is a determiner

Table 4: Sentence with Determiner.

In the above example, *Every* is a determiner and when considering it, our system will be able to extract the information that there are *28 lunchboxes*.

Existential/Expletive: Since, Existentials/Expletives indicate the existence or presence of something they are an important Part of Speech for our system. Also, when the sentences are simplified, the existentials are added to the fragments which don't have them. Refer to the next section for more details.

Noun: Nouns are important for answering the arithmetic word problems. Each problem text has some Nouns in form of a subject *(acting entity)* or an object *(Entity that is acted upon by the subject)*. All the nouns are extracted from the simplified sentence and stored as a list.

Preposition: Prepositions are helpful to keep track of location or time. Also when simplifying sentences based on conjunctions we use this representation to add to the sentences which are missing prepositions. Refee to the next section for more details and example.

Verb: Verbs are important for out system to predict the type of action taken by the subject. Determining the kind of action helps to predict an operation. More details on this in Section Features.

WHAdverb: Existence of a WHAdverb mostly indicates the beginning of a question. Hence, having the wbadverb in our representation helps us in the classification process. Refer to the Section Features.

4 Sentence Simplification and Problem Decomposition

Sentences in an arithmetic word problem are sometimes complex. Hence, its difficult to extract information from such sentences. More than extracting the information it is difficult to predict the impact of the sentence on the result.

The second sentence in the above example is complex for a machine learning algorithm. It has addition and subtraction operation in a single sentence. Our idea is to simplify the sentence to multiple simple sentences so that each simplified sentence has a single operation. Below are the simplified

Example 2:
Henry had 11 dollars. For his birthday he got 18 more dollars but spent 10 on a new game. How much money does he have now?

Table 5: Example Arithmetic Word Problem.

sentences for the second sentence in the above example:

For his birthday he got 18 more dollars but spent 10 on a new game.
For his birthday he got 18 more dollars.
He spent 10 dollars on a new game.

Table 6: Simplified Sentences.

We create a mapping for each sentence in the problem text to its simplified sentences by extracting the relational dependencies for each sentence from the stanford dependency parser[1]. Multiple sentences are created based on the dependencies of the actual sentence. Currenly, our simplification system simplifies sentences based on conjunctions and commas. There are certain rules when simplifying the sentence.

4.1 Rules for simplifying sentences based on conjunctions.

When a conjunction is encountered, our simplification system attempts to create two simplified sentences from the actual sentence. The first sentence is the part before the conjunction while the second sentence is the part after the conjunction.

But, after the split there may be some words which would be required in the second sentence. Consider the example in Table 4:

Example Sentence.
The school cafeteria ordered 42 red apples and 7 green apples for students lunches.

In the sentence above, the split based on conjunction 'and' will result in the first sentence with a Noun while the second will not have a Noun and a Verb. Hence, there are some rules for adding words to the second simplified sentence.

[1]http://stanfordnlp.github.io/CoreNLP/

4.1.1 Rules for adding words to second simplified sentence.

1. If the first sentence starts with an Expletive/Existential and the second does not, add the Expletive/Existential to the second sentence as well.

2. If the first sentence starts with an Expletive/Existential and has a Verb after the Expletive, If the second sentence does not have any of the Expletive/Existential and Verb, add them to the second sentence. Consider the below example:

There were 2 siamese cats and 4 house cats.
There were 2 siamese cats.
There were 2 house cats.

Table 7: Example sentence for this rule.

The expletive and verb were added to second sentence based on the simplification rule mentioned above.

3. If the first sentence starts with a Noun, and if the second sentence starts with a Verb, the Noun from the first sentence will be added to the second.

Joan ate 2 oranges and threw 3 apples.
Joan ate 2 oranges.
Joan threw 3 apples.

Table 8: Example sentence for this rule.

4. If the first sentence starts with a Noun and the second sentence has a *Noun Verb* pattern, do nothing.

Tom has 9 yellow balloons and Sara has 8 yellow balloons.
Tom has 9 yellow balloons.
Sara has 8 yellow balloons.

Table 9: Example sentence for this rule.

No words from the first sentence were added to the second since the second sentence had the *Noun Verb (Sara has)* pattern.

5. If the second sentence contains a preposition at the end and the first sentence does not have a preposition, the preposition from the second sentence will be added to the first. Consider the below example:

Joan found 6 seashells and Jessica found 8 seashells on the beach.
Joan found 6 seashells on the beach.
Jessica found 8 seashells on the beach .

Table 10: Example sentence for this rule.

After splitting the sentence based on *and* the preposition and the words after it *on the beach* were added to the first sentence.

6. Based on the output by the dependency parser and our rules, there might be some words which might not have been identified. But we still need those words in the simplified sentences.

7. Therefore, the sentence simplification system identifies all the words which were not identified by our process. The words which appear before the conjunction are added to the first sentence at the right index. If the words appear after the conjunction, they are added to the second sentence.

He went to the orchard and picked peaches to stock up .
Tom has 9 yellow balloons.
Sara has 8 yellow balloons.

Table 11: Example sentence for this rule.

References

Hosseini, M. J., Hajishirzi, H., Etzioni, O., and Kushman, N. (2014). Learning to solve arithmetic word problems with verb categorization. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 523–533. ACL.

Kushman, N., Artzi, Y., Zettlemoyer, L., and Barzilay, R. (2014). Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland. Association for Computational Linguistics.

Roy, S. and Roth, D. (2015). Solving general arithmetic word problems. In Mrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *EMNLP*, pages 1743–1752. The Association for Computational Linguistics.

Roy, S., Vieira, T., and Roth, D. (2015). Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics*, 3:1–13.