Automatically Solving Word Algebra Problems with Structured Prediction Energy Networks

Arpit Jain
jain@umass.edu

Gota Gando
ggando@umass.edu

Krishna Prasad Sankaranarayanan
ksankaranara@umass.edu

Nikhil Yadav
nikhilyadav@umass.edu

Abstract

To solve algebra word problems automatically, typical approaches find a transformation from the given word problem into a set of equations which correctly represents the input word problem. To approach this task, we apply structured prediction energy networks (SPENs), which is an energy based model where we can inject the structure knowledge of the task. We compare SPENs to a baseline of a simple feed-forward neural network model to determine the effectiveness of modeling structural dependencies in this problem.

1 Introduction

Algebra word problems in general describe some particular world situation and pose a question about it. In this project, we only consider the problems where we can use a system of equations to solve the questions. Figure 1 shows one example of such problems. In this case, we can express the problem as a set of algebraic equations that descrbe the mathematical relationship of entities in the problem: for example $m = 4 \times n$ and $m - 4 = 6 \times (n - 4)$ where m and n represent the age of Maria and Kate, respectively. To answer the question, we need to solve these equations and get the solution. However, we focus on finding a mapping from the word problem into an equation system in this project, as there are many automated solvers for this process.

Maria is now four times as old as Kate. Four years ago, Maria was six times as old as Kate. Find their ages now.

Figure 1: An example of word algebra problem.

1.1 Structured Prediction

If we consider each character of the output equations as a random variable, then this task can be considered as a structured prediction task, as every output random variable would be dependent on some other variables. Furthermore, as both the input and output are sequences, we can also interpret this task as a sequence-to-sequnce task.

In many machine learning tasks, we try to predict $\mathbf{y}$ given an input $\mathbf{x}$. In some cases it is sufficient to use a discriminative function F to predict the output such as $\mathbf{y} = F(\mathbf{x})$. However, this model fails to model the structure of the output and the interactions among output components and may perform poor on more complex tasks. We can instead use *an energy function* E that both depends on $\mathbf{x}$ and $\mathbf{y}$ to model such structure and seek the optimal $\mathbf{y}$ that minimizes the energy:

$$\hat{\mathbf{y}} = argmin_{\mathbf{y}} E(\mathbf{y}, \mathbf{x}) \tag{1}$$

Energy function can be seen as a scoring function that returns how compatible $\mathbf{x}$ and $\mathbf{y}$ are, or how likely they occur together. One thing we need to note here is that there is no guarantee of the returning value of energy functions will be in the range of $\{0, 1\}$. One common way to address this problem and calibrate the energy is putting it through the Gibbs distribution.

$$P(Y|X) = \frac{\exp -\beta E(Y, X)}{\int_{y \in (Y)} \exp -\beta E(y, X)} \tag{2}$$

However, it should be noted that this transformation is only possible when computing the integral term $\int_{y \in (Y)} \exp -\beta E(y, X)$ is tractable.

2 Related Works

Solving word albegra problems automatically is a long-standing task in natural language processing (NLP). ...

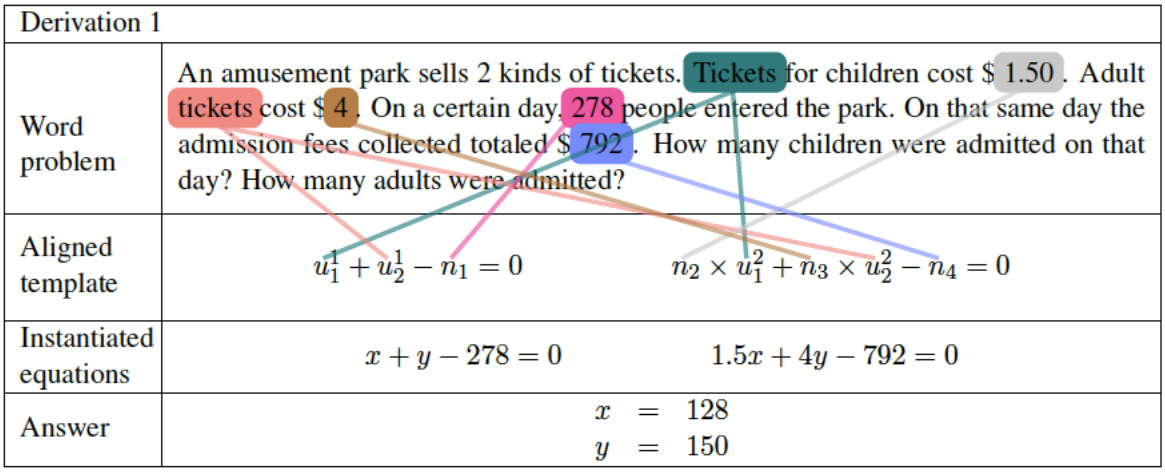

Figure 2: The structure of the baseline model. Cited from (Kushman et al., 2014)

3 Problem Definition

Following (Kushman et al., 2014), we define a transformation of word problems to equations as a two step process. First, a template is selected to define the structure of the target equation system. Second, the selected template is instantiated with actual texts for unknowns and values for coefficients. Figure 2 shows an example of template selection and instantiation. Each template has u slots that represent unknown variables, and n slots for coefficients of unknowns.

3.1 Derivation

In this section, we review the formulation described in (Kushman et al., 2014) and (Upadhyay et al., 2016). We denote the input word problem as $\mathbf{x}$, and the output as $\mathbf{y}$. We call the output as a derivation, which consists of a template T and an alignment A. Therefore we can express the output as $\mathbf{y} = (T, A)$. More specifically, template T is a set of equations $= \{t_0, ..., t_l\}$, and an alignment A is defined as a set of pairs (w, s), where where w is a token in $\mathbf{x}$ and s is a slot instance in T.

4 Approach

4.1 Baseline model

We plan to use two baseline models. One is a regular feed-forward neural network that predicts the output equations directly from the input, interpreting the task as a black-box sequence-to-sequence task. Another baseline is also a neural network that jointly predicts derivation of templates and its alignments. In both cases, first the input text problem is put through a word embedding layer and a vector is assigned to each word. The first baseline then predicts the output directly after forwarding through several intermediate layers. The second baseline model on the other hand, tries to predict the template structure and its alignments at the same time. This is performed by constructing a derivation vector where the first element denotes the index of the predicted template and the other elements denote the textual or integer values of each unknowns / coefficients in the selected template.

4.2 SPEN

Structured Prediction Energy Networks (SPENs) (Belanger and McCallum, 2015)(Belanger et al., 2017) are one of the energy-based models (EBMs) that performs structured prediction, such as conditional random fields (CRFs), structured perceptrons, and structured SVMs (SSVMs). There are two main differences betwen SPEN and other enegy-based models. First, SPENs can learn nonlinear energy functions, as it is parameterized by a deep neural network. Similar to other EBMs, SPEN also tries to predict y that minimizes the energy function. While EBMs listed above usually assume a restricted graphical structure such as a chain or a tree and consider a linear energy function, SPEN instead considers a general energy function and assumes the non-convexity. Then, it approximates the optimal $\mathbf{y}$ with gradient descent:

$$\bar{\mathbf{y}} = argmin_{\bar{\mathbf{y}}} E(\bar{\mathbf{y}}, \mathbf{x}) \tag{3}$$

$$\mathbf{y}^{t+1} = \mathbf{y}^t - \eta \frac{\partial E}{\partial \mathbf{y^t}} \tag{4}$$

Additionally, SPENs are much more efficient at the inferene stage where the model predicts a candidate output $\bar{\mathbf{y}}$, as SPENs use gradient descent to approximate an optimal $\bar{\mathbf{y}}$ instead of using computationally expensive searching methods such as viterbi algorithm or beam search.

4.3 Architecture of SPEN

Although SPEN can take general energy functions, in this work we consider the energy function consists of the global energy and the sum of local energies as follows:

$$E_x^{local}(\bar{y}) = \sum_{i=1}^{L} \bar{y}_i b_i^T F(x) \quad (5)$$

$$E_x^{global}(\bar{y}) = c_2^T g(C_1 \bar{y}) \quad (6)$$

where g is a non-linearity function, each b_i is a vector of parameters for each label, and the product $C_1\bar{y}$ is a set of learned affine (linear + bias) measurements of the output.

We give the same representation of y as described in the baseline section to this energy network.

5 Evaluation

Following the existing works, we evaluate the model's performance with the overall accuracy on the ALG-514 and DRAW datasets. The predicted equation system is marked as correct if all the characters of the output exactly matches the ground truth equations. We also plan to evaluate the accuracy on several subsets of word problems to further analyze problems the model can perform well.

6 Datasets

We plan to use two datasets; the ALG-514 dataset proposed by (Kushman et al., 2014) which contains 514 word problems, and the DRAW dataset provided by Microsoft that contains about 1000 independent examples. One issue we are currently aware of these datasets is that ,while they include template structure of output equations for each problem, they do not provide the mapping of unknowns to text values. We plan to manually annotate such alignment information on these datasets so that we can fully make use of the template structure and its alignments.

7 Scope

Although it might be interesting to explore how we can improve the model itself, we limit the scope of this project to applying and evaluating the baseline MLP and SPEN model. Application of the SPEN includes designing structured representation of the output variable y. We are also planning to utilize the "implicit" data samples described in (Upadhyay et al., 2016), which only contain the solutions of equations and not equation themselves if we have time after evaluating the baselines and the SPEN model. In this approach, we hope to jointly learn from data samples with equation labels and without such labels.

References

David Belanger and Andrew McCallum. 2015. Structured prediction energy networks. *CoRR*, abs/1511.06350.

David Belanger, Bishan Yang, and Andrew McCallum. 2017. End-to-end learning for structured prediction energy networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 429–439.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *TACL*, 3:585–597.

Nate Kushman, Luke S. Zettlemoyer, Regina Barzilay, and Yoav Artzi. 2014. Learning to automatically solve algebra word problems. In *ACL*.

Yann Lecun, Sumit Chopra, Raia Hadsell, Fu Jie Huang, G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar (eds. 2006. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press.

Arindam Mitra and Chitta Baral. 2016. *Learning to use formulas to solve simple arithmetic problems*, volume 4. ACL.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *CoRR*, abs/1608.01413.

Shyam Upadhyay, Ming-Wei Chang, Kai-Wei Chang, and Wen-tau Yih. 2016. Learning from explicit and implicit supervision jointly for algebra wo In *EMNLP*.