Uncertainty-Based Joint Training for semi-supervised Math Word Problem

Anonymous ACL submission

Abstract

Math word problems (MWPs) convert natural math corpus into structured equation forms. Data sparsity is one of the main obstacles for math word understanding problem due to the high cost of human annotation efforts. However, existing work mainly start from the supervised learning perspective, making the low-resource scenario under explored. In this paper, we are the first to incorporate semi-supervised learning (SSL) framework into MWPs. We propose an uncertainty-aware unlabeled data selection strategies, which can access to reliable samples and increase the model capacity gradually. Besides, to improve the quality of pseudo equations, we incorporate two indirect supervision signals considering the semantic consistency property and grammar format constraints of generated equations. Experimental results on two benchmark MWPs datasets across different ratio of unlabeled data verify the effectiveness and generalization ability of our proposed method.

1 Introduction

Developing computer system to automatically solve math word problems (MWPs) dates back to 1960s (Bobrow, 1964) and is an important task in natural language understanding. It maps a textual description to a logical equation expression. A logical equation is machine-understandable and can be executed directly for a numerical answer. To achieve that, we need to identify the relevant quantities from the text and determine the correct operators as well as computation order among these numbers. One example of MWPs is shown in Figure 1. It is required to first understand the semantic meanings of quantities like "spend total, dollar, 288", and induce operators reflecting their relations. Then the logic form should be collected and arranged in the mathematical rules, e.g., each valid equation can be parsed into a complete binary tree

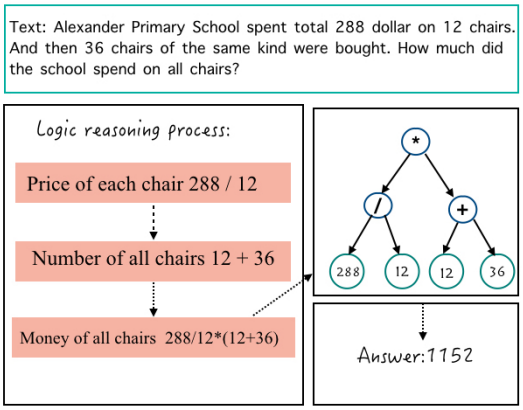

Figure 1: An example of math word problems.

with numeric on the leaf. We can use this math grammar format in this paper.

Earlier works (Koncel-Kedziorski et al., 2016; Hosseini et al., 2014; Roy and Roth, 2018) depended upon manually-designed template annotations to train a successful model. However, their paradigm of designing and obtaining hand-crafted features is difficult to generalize to larger and more complex datasets. Recent researchers introduced deep learning techniques for better language understanding and logic reasoning (Wang et al., 2017; Xie and Sun, 2019; Liu et al., 2019). Although they are able to learn more expressive representations with the benefit of deep neural networks, the need for a large amount of supervision persists. Recently, (Hong et al., 2020) proposed a learning-by-fixing framework in weakly supervised setting. However, they still need mathematical results as a side-supervision. The collection of data annotations for MWPs is a labor-intensive and time-consuming task, as calculating math equations is unfriendly for human annotators. To make it worse, there are many domain knowledge and concepts in MWPs which limits the effect of general knowledge transfer algorithms in MWPs language understanding.

To reduce human intervention and boost data utilization, it is vital to design a new algorithm that can effectively utilize unlabeled math corpus. An intuitive idea is to incorporate semi-supervised learning (SSL) framework into MWPs. We can first train the model on small-scale labeled data and then apply the trained base model on unlabeled data to generate pseudo equations. These obtained pseudo-labeled data can augment the training data and be utilized to retrain the MWPs model. However, there are two challenges in incorporating SSL into MWPs:

– How to select reliable samples from unlabeled data pools?

– How to improve the quality of obtained pseudo equations?

Vanilla SSL mechanism randomly pick up unlabeled samples in the pseudo labeling process, which will inevitably introduce noise since it ignores the reliability of selected data. An intuitive remediation is to feed data in an easy to hard procedure and enhance the model learning capacity gradually. Recently, the so-called curriculum learning achieves promising results in machine translation (Zhou et al., 2020), emotion generation task (Shen and Feng, 2020) and etc. They often measure sample difficulty from human linguistic knowledge perspective, like sentence length (Zhou et al., 2020), rare word numbers (Shen and Feng, 2020). However, such pre-defined data difficulty measurement can hardly be applied to MWPs. Intuitive criteria like equation length does not square well with the reasoning complexity of math problems, making it non-trivial to select reliable samples. Besides, no annotation supervision is available in the pseudo labeling process. Wrongly generated equations will propagate errors to the following retraining phrase. Furthermore, math equation follows strict grammar structure, making it even more challenging to generate high-quality pseudo equations.

To mitigate the above two problems, we propose an **U**ncertainty aware **S**emi-supervised learning for **M**ath word problems (USM). We propose an uncertainty-based measurement for reliable data selection. Furthermore, we introduce two indirect supervision signals to regularize the validity of pseudo equation. To be more specific, inspired by recent advance in Bayesian deep learning (Bernardo and Smith, 2009; Gal and Ghahramani, 2016) to obtain uncertainty estimation, we take the model uncertainty as a data selection measurement. Besides, although there is no annotation for pseudo equation generation procedure, we incorporate two additional indirect supervised losses considering MWPs semantic consistency and grammar format constraints. We propose a question paraphrasing task to make the math corpus and pseudo equations to reconstruct initial questions. Meanwhile, we design an equation grammar checker to restrict the the pseudo equation following their valid grammar requirement.

Contributions The main contributions of this paper are summarized as follows: 1) To the best of our knowledge, we are the first to investigate the semi-supervised MWPs in which unlabeled data has no supervision. 2) We further use model uncertainty to assess data quality and select reliable pseudo-labeling data from unlabeled pools to regularize the learning process of DNN models on the MWPs. 3) To better serve unlabeled samples with supervision signals, we explore to utilize a question paraphrasing mechanism to ensure their semantic meaning alignment and design a specific equation grammar checker reward to meet their grammar requirement. 4) We conduct extensive experiments on Math23K and MAWPS dataset. The results show the effectiveness and generalization of the proposed method.

2 Methodology

In this section, we propose a systematic USM framework to deal with the semi-supervised math word problem. One of the typical semi-supervised learning schemes is to train on the labeled data first and then deploy on the unlabeled data to get pseudo labels. The new synthetic data is utilized as data augmentation to retrain the model. There are two challenges under this setting: 1) How to select reliable unlabeled samples for augmentation? MWPs is a reasoning task and the logic difficulty is hard to be represented. 2) How to design indirect supervision signals for pseudo equation generation process? There are no annotations for unlabeled corpus, and noise from pseudo equations could accumulate and even degrade the training stability and efficiency.

USM provides a systemic solution consisting two main sub-steps, uncertainty-aware data selection and pseudo equation enhancement. We first present a brief introduction to the basic framework of USM, including notations and setting. Then we

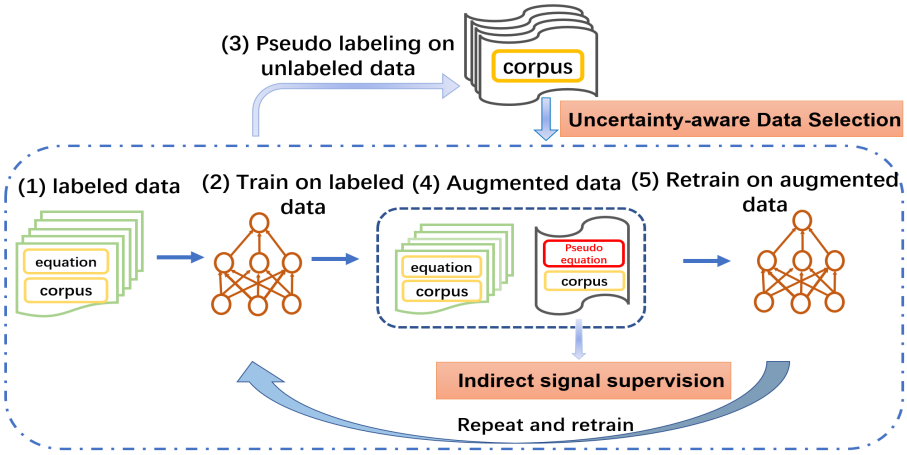

Figure 2: Overview of the proposed approach. We first train on the labeled data, and apply the trained model on unlabeled data to select reliable samples and generated pseudo equations. Then we retrain the model on augmented data. In this paper, we propose a uncertainty-based data selection strategy and introduce two additional indirect supervision signals for pseudo equation generation.

describe the detail of the main techniques.

2.1 Overall framework

Consider $D_l = \{(x_i, y_i)\}_{i=1}^{N}$ to be a set of N labeled documents (instances) with y_i being the equation for x_i. Each x_i consists of a sequence of word tokens and numerical values. The goal of math word problem is to map x_i to a valid and correct equation y_i. Specifically, there are various of M unlabeled corpus $D_u = \{x_u\}_{u=1}^{M}$ which should be further leveraged in the semi-supervised setting.

In this paper, we first train a model on labeled data D_l with cross entropy loss. Then we apply the trained teacher model on unlabeled instances D_u to obtain synthetic labeled pairs. Specifically, we calculate model uncertainty for each synthetic samples $\{x_u, \hat{y}_u\}$, and select reliable samples based on the calculated confidence. Further more, instead of leveraging the synthetic labels directly, we introduce two additional indirect supervision signals to force the validity of the generated equations. Then, iterating the process by putting back the student as a teacher to generate new pseudo labels and train a new student. The three step process can iterate until convergence. A schematic description of the overall framework is shown in Figure 2.

2.2 Uncertainty-aware Data Selection

A straightforward selection method is to randomly select pseudo-labeled samples at each time step. It is evident that such strategy could suffer from the noises brought by the teacher model, especially on difficult samples. A natural idea is to organize math word problems in a meaningful order which illustrates increasing concepts diversity and equation complexity, and train the model in an easy-to-hard manner. Prior works access task difficulty/uncertainty in sentence length (Zhou et al., 2020), rare word numbers (Shen and Feng, 2020) and so-called curriculum learning strategies achieve great success recently (Miller and Seller, 1985). However, it is non-trivial to define the model confidence in MWPs. Corpus length does not square well to logic reason ability and training loss may be invalid since long simple equations will accumulate large training loss.

To identify reliable samples, we propose to leverage model uncertainty (Gal and Ghahramani, 2016) as the selection criterion for pseudo-labeled data. Model Uncertainty is also known as epistemic uncertainty (Hofer et al., 2002), which can be used to quantify the confidence of model towards its prediction. With an unlabeled math corpus x_u, the synthetic equation $\hat{y}_u$, the translation probability under the trained teacher model can be represented as:

$$p(\hat{y}_u) = \int_{\theta} p(\hat{y}_u | f(x_u; \theta)) p(\theta | x_u, \hat{y}_u) d\theta. \quad (1)$$

Generally, we mainly ignore model uncertainty term $p(\theta|x_u, \hat{y}_u)d\theta$, which reflects our confidence about model parameters. While Bayesian neural network aims to find the posterior distribution over the model parameters $p(\theta|x_u, \hat{y}_u)$. This formulation requires us to average all possible model weights which is intractable realistic practice. For reasons of computational efficiency, we adopt the

widely used Monte Carlo Dropout (Gal and Ghahramani, 2016) to get the equation-level uncertainty.

Given the current unlabeled data x_u, the trained teacher model makes its prediction $\hat{y}_u$ via a standard decoding process. To further estimate model uncertainty about current prediction, we randomly deactivate part of neurons and perform T pass of forward propagation. For every pass, we recalculate translation probabilities while keeping x_u fixed. Eventually, we obtain T samples over model parameters $\{\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_T\}$ and the translation probabilities. Intuitively, low variance indicates that the model is confident about its prediction. Given T samples $\{p(\hat{y}_u|x_u, \hat{\theta}_t)_{t=1}^T\}$, the equation-level translation probability can be represented as:

$$\mathbb{E}[p(\hat{y}_u|x_u, \hat{\theta})] \approx \frac{1}{T}\sum_{t=1}^{T} p(\hat{y}_u|x_u, \hat{\theta}_t). \quad (2)$$

The variance of equation-level uncertainty can be represented as :

$$\mathrm{Var}[p(\hat{y}_u|x_u, \hat{\theta})] \approx \frac{1}{T}\sum_{t=1}^{T} p(\hat{y}_u|x_u, \hat{\theta}_t)^2 - \mathbb{E}[p(\hat{y}_u|x_u, \hat{\theta})]^2, \quad (3)$$

where $\mathrm{Var}[p(\hat{y}_u|x_u, \hat{\theta})]$ represents model uncertainty calculated via math corpus x_u. In this paper, we gradually select unlabeled data which uncertainty score is less than a fine-tuned threshold β.

2.3 Indirect Signals for Pseudo Equation Generation

Besides selecting reliable unlabeled samples, we come to the problem of designing auxiliary supervision signals to improve the quality of generated pseudo equation. Although there is no annotation for unlabeled corpus, we claim there are two specific properties in MWPs:

- Equations have strong correlations with the math corpus semantic information.
- Equations should follow mathematical grammar format and operator precedence.

Based on the two intuitions, we introduce question paraphrasing to hold semantic consistency and design a equation grammar check reward to keep a valid equation logic form following its grammar format constraints.

2.3.1 Question Paraphrasing.

Paraphrasing aims to perform semantic consistency and try to bridge the gap between math context understanding and logical equations. A straightforward implementation is to deploy the trained teacher model on unlabeled corpus x_u and generate pseudo-labeled data $\hat{y}_u$. $\hat{y}_u$ is used to reconstruct

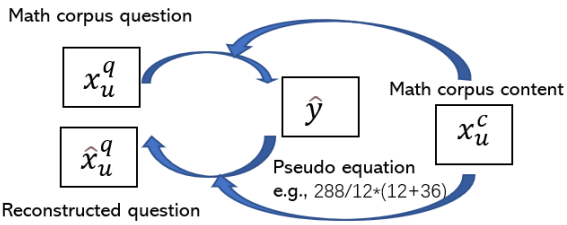

Figure 3: Overview of the math question paraphrasing. To keep the semantic information of generated equations, we leverage the math corpus content and pseudo questions to reconstruct the math question. For each $\hat{y}_u$, we use its reversed quantity mapping y_{qm} in our experiments.

the corpus information x_u. However, there is a semantic drop in the $\hat{y}_u \rightarrow x_u$ reconstruction direction. First, it is an ill-posed problem to predict math corpus from equations since the same equation can correspond to various of math corpus. Besides, the logical equation itself contains no semantic information, let alone reconstruct it. Facing with this difficulty, we observe that each equation usually has a strong correlation with the question while not with the corpus content, as we mentioned in Fig.1.

Motivated by this, we split each unlabeled math corpus x_u to math content x_u^c and math question x_u^q. We first deploy the trained model on unlabeled corpus x_u to obtain the pseudo equation $\hat{y}_u$. Then we leverage the math content x_u^c and the pseudo labeled equation $\hat{y}$ to reconstruct the math question x_u^q. This implementation would mitigate the one-to-many problem since it enhances the correlation between generated equations and corresponding questions. And the incorporating of math content avoids the loss of semantic information at the most extent. We can exploit the attention-based Encoder-Decoder architecture to build this model. Figure 3 illustrates the question paraphrasing process.

Given the concatenation x_u of math content word x_u^c and reversed quantity mapping y_{qm} (we will explain later), each word is mapped to a fixed dimensional vector by a word embedding function $\phi(\cdot)$ and then fed into a bidirectional LSTM (Huang et al., 2015). The hidden vectors can be recursively calculated at each time step:

$$\begin{aligned}
\overleftarrow{\boldsymbol{h_i}} &= f_{LSTM}(\phi(x_i), \overleftarrow{\boldsymbol{h}}_{\boldsymbol{i-1}}),\ i = 1, 2, ..., |x| \\
\overrightarrow{\boldsymbol{h_i}} &= f_{LSTM}(\phi(x_i), \overrightarrow{\boldsymbol{h}}_{\boldsymbol{i-1}}),\ i = 1, 2, ..., |x| \\
\boldsymbol{h_i} &= [\overleftarrow{\boldsymbol{h_i}}; \overrightarrow{\boldsymbol{h_i}}],
\end{aligned} \quad (4)$$

where $\boldsymbol{h_i}$ denotes the hidden states, $|x|$ is the number of input tokens, $[;]$ denotes the vector concate-

nation and f_{LSTM} is the LSTM function. Decoder is an unidirectional LSTM with attention mechanism. The hidden state at t-th time $\boldsymbol{s_t}$ is calculated by $\boldsymbol{s_i} = f_{LSTM}(\phi(y_{i-1}), \overleftarrow{\boldsymbol{s}}_{\boldsymbol{i-1}})$ with initialization $\boldsymbol{s_0} = \overleftarrow{\boldsymbol{h_1}}$. The attention weight for the current step t of the decoder, with the i-th step in the encoder is cauculated by $a_t = a_i^t = \frac{exp(u_i^t)}{\sum_{i=1}^{|x|} exp(u_j^t)}$ and

$$\begin{aligned} \boldsymbol{u_i^t} &= \boldsymbol{v^T}\, tanh\, (\boldsymbol{W_1 h_i} + \boldsymbol{W_2 s_i} + \boldsymbol{b_a}) \\ \boldsymbol{c_t} &= \sum_{i=1}^{|x|} a_i^t \boldsymbol{h_i} \\ P_{gen} &= softmax(\boldsymbol{W_0}[\boldsymbol{s_t}; \boldsymbol{c_t}] + \boldsymbol{b_0}), \end{aligned} \tag{5}$$

where $\boldsymbol{W_1}$, $\boldsymbol{W_2}$, $\boldsymbol{v}$, $\boldsymbol{b_a}$ are trainable parameters. $\boldsymbol{W_0}$ and b_0 mapping the concatenation of hidden state to the output vocabulary size. In the end, we apply cross-entropy loss to reconstruct the math questions:

$$\mathcal{L}^{rec} = -\sum_{i=1}^{M} x_{u_i}^q \dot{p}_{gen}(\hat{x}_{u_i}^q | \hat{x}_{u_{i-1}}^q, [x_u^c; \hat{y}_{qm}]). \tag{6}$$

Here we explain the implementation of ***reversed quantity mapping*** $\hat{y}_{qm}$. Quantity itself has no semantic meaning, and will be ignored if we only use numerical representation. Hence, we reversely map every possible equation numeric to its corresponding noun phrase before question generation. Since each numeric quantity may have multiple aliases in the real world, e.g. 2 can corresponds to 2 hour and can also associated with 2 kilometers. We parse the initial text and consider the nouns related with the numeric quantity in the dependency trees as the quantity expression. We also map each operator "+, -, *, /" as "add, subtract, times, divide". Finally, the mapped equation $\hat{y}_{qm}$ is concatenated with math content x_u^c and are fed into encoder as input x_u.

2.3.2 Equation grammar checker

Besides enforcing semantic consistency loss to pseudo equations, we further consider equation grammar constraints and introduce additional indirect supervision signals. (Wang et al., 2017) is the first to claim that wrong equations will be generated if we ignore it structure constraints, i.e., '3++8' or '(677+'. Inspired by this, its following works (Wang et al., 2018, 2019; Liu et al., 2019) aim at incorporating such structure prior information into model architecture design. Here we share a similar idea but for pseudo equation quality enhancement. Specifically, we check the equation format validity with the following loss:

$$\mathcal{L}^{gra}(\hat{y}) = \text{grammar\_error\_indicator}(\hat{y}). \tag{7}$$

For each valid pseudo equation, we assume it should follow four requirements: 1) if last token are in $\{+, -, *, /\}$, next token will not be in $\{+, -, *, /,)\}$; 2) if last token is a numeric, next token will not in $\{\}$; 3) if last token is a $\{(\}$, next token will not in $\{(,), +, -, *, /,)\}$; 4) if last token is in $\{)\}$, last token will not in $\{(,)\}$. The above indicator returns 1 when y has no error at the grammar format levels, and returns 0 otherwise. Since feedback reward is non-differentiable here, reinforcement learning algorithm (Kaelbling et al., 1996) based on policy gradient (Silver et al., 2014) is applied for optimization:

$$\begin{aligned} \nabla_\theta \mathbb{E}(R) =& \nabla_\theta \sum_k P(\hat{y}_k|x;\theta) R_k \\ =& \sum_k P(y_k^{'}|x;\theta) R_k \nabla_\theta \log(P(\hat{y}_k|x;\theta)) \\ \simeq& \sum_k R_k \log(P(\hat{y}_k|x;\theta)). \end{aligned} \tag{8}$$

For each selected unlabeled sample, we generated K possible pseudo equations $\hat{y}_1$, $\hat{y}_2$,...,$\hat{y}_K$, where K is the number of beam size, and R_k is given by $R_k = \alpha\mathcal{L}^{rec} + (1-\alpha)\mathcal{L}^{gra}$. The pseudo algorithm of our framework is shown in Alg.1 in appendix.

3 Experiment

Dataset We evaluate our method on the widely used Math23K (Wang et al., 2017) and MAWPS dataset (Koncel-Kedziorski et al., 2015). We set up an experiment to evaluate semi-supervised MWPs with a varying amount of labelled training data $(25\%, 50\%, 75\%)$, with the rest being unlabeled.

Baselines We compare our approach with extensive of representative baselines: (1) **DNS** (Wang et al., 2017): A vanilla Seq2Seq with bidirectional Long Short Memory model. (2) **Math-EN** (Wang et al., 2018): An ensemble model based on BiLSTM and transformer with equation normalization (EN). (3) **S-aligned** (Chiang and Chen, 2018): Neural symbolic based model which utilizes the stack to generate associated equations. (4) **Group-ATT** (Li et al., 2019): A variant of transformer based on group attention. (5) **GTS** (Xie and Sun, 2019): A goal driven seq2tree based model. (6) **Graph2Tree** (Zhang et al., 2020): An extension of **GTS** with graph-transformer encoder. Our proposed framework is model-agnostic, we conduct experiments

Table 1: Equation accuracy and answer accuracy over all baselines on Math23K dataset.

Method		0.25		0.5		0.75		1	
		E-acc	A-acc	E-acc	A-acc	E-acc	A-acc	E-acc	A-acc
Supervised	S-Aligned	41.17	47.32	49.79	55.21	57.74	61.93	61.43	65.8
	Group-ATT	43.45	48.67	52.09	57.15	60.42	65.58	63.98	69.2
	Graph2tree	50.41	58.79	57.21	67.97	62.68	72.77	63.82	75.38
	DNS	41.03	43.83	48.07	54.74	55.53	61.32	59.34	66.15
	GTS	49.22	56.12	57.32	66.75	59.69	70.68	64.59	74.32
	BERT	54.30	62.15	61.14	72.37	67.33	78.71	70.94	81.80
Semi-supervised	DNS-USM	44.69	48.91	52.14	56.90	58.20	63.65	-	-
	GTS-USM	52.57	59.75	60.14	70.91	61.03	72.58	-	-
	BERT-USM	57.68	66.29	63.15	74.27	68.99	80.04	-	-

Table 2: Equation accuracy and answer accuracy over all baselines on MAWPS dataset.

Method		0.25		0.5		0.75		1	
		E-acc	A-acc	E-acc	A-acc	E-acc	A-acc	E-acc	A-acc
Supervised	S-Aligned	41.17	53.76	65.45	66.23	69.10	69.61	72.73	73.51
	Group-ATT	53.24	55.58	67.71	68.57	72.99	73.77	76.10	76.62
	Graph2tree	61.45	62.76	76.82	77.08	79.16	80.21	83.07	83.33
	DNS	52.73	53.25	66.94	67.71	72.21	72.92	76.62	77.14
	GTS	60.41	61.97	75.26	76.04	78.90	79.17	81.25	82.29
	BERT	66.41	67.18	77.14	77.60	81.18	81.18	83.63	84.11
Semi-Supervised	DNS-USM	57.14	57.92	69.61	70.39	74.29	75.00	-	-
	GTS-USM	64.84	65.89	77.34	78.38	81.25	81.51	-	-
	BERT-USM	69.87	70.38	78.70	79.48	82.81	83.85	-	-

on the top of three general model architecture: DNA, GTS and BERT (Devlin et al., 2018).

Implement Details Word embedding dimension is set to 128 in our experiments. The dimension of hidden state for all the other layers are set to 512. Batch size and dropout rates are set to 64 and 0.5. Also, we use a beam size of 5 in beam search. Our model are trained with 80 epoches for GTS and BERT model, and trained with 200 epoches for DNS model on the two dataset. We use Adam optimizer (Kingma and Ba, 2015) with initial learning rate 0.001. For the uncertainty-aware data selection procedure, we set the number of forwarding times as 16. Besides, we set the uncertainty selection threshold as the mean uncertainty over whole dataset times 0.2. For the indirect signal enhancement part, we set the trade-off value of α as 0.5.

3.1 Overall Results

We perform USM baselines on Math23K and MAWPS dataset separately. The experiments results are shown in Table 1 and Table 2 separately. Since baseline models are mainly designed for supervised learning, so we apply their model on the splitting labeled data. We further select three distinctive methods DNS, GTS and BERT and apply them with our USM framework. As shown in Table 2 and Table 3, our proposed USM method consistently outperforms all the baselines for all the datasets w.r.t. different ratio of labeled data. For example, GTS achieves 57.32% equation accuracy with 50% labeled data, and the result can increase to 60.14% when applying on GTS-USM model. We ascribe the reason to that the various of selected pseudo-labeling math corpus enhance the model capacity in a data-driven way, and pseudo-labeling samples provide additional signals to regularize the math equation generation process. Different from previous works which design more complex model architecture or annotate more label data with high human annotation cost, leveraging unlabeled corpus effectively can also boost the performance of math equation generation.

Besides, our proposed method is both general and effective. It can be easily incorporated into various model architectures, like seq2seq model (DNS), graph2tree model (GTS) and current widely used BERT model. Meanwhile, it achieves consistency promising results on these model architecture. Compared with DNS, GTS and BERT model, our proposed USM framework achieve relative 4%, 3% and 2% improvements with 50% labeled samples on math23K dataset. The promising results demonstrates the generalization ability and effectiveness of our proposed semi-supervised framework.

In addition, USM achieves promising results than all baselines, especially in low-resource scenarios. It is promising that BERT-USM achieves

3.46% improvements over BERT model with 25% labeled data in MAWPS dataset. Especially, GTS-USM obtain 77.34% equation accuracy on 50% labeled data which is approximating to 78.90% of GTS model achieves with 75% labeled data. The satisfactory results demonstrates the viability of our proposed semi-supervised learning framework.

3.2 Ablation Study

To better understand the performance contributed by each proposed component, we perform a series of ablation tests by removing components one by one. We first investigate different unlabeled sample selection strategies. Then we discuss the effectiveness of two indirect signals. The results are demonstrated in Table and Table respectively.

3.2.1 Effects of Data Selection.

To show the influence of different selection strategies, we investigate the following methods on DNS-USM and GTS-USM model:

- USM: Uncertainty-aware semi-supervised Learning framework for MWPs (ours).
- CSM: Curriculum based framework for MWPs. We select sample from a easy to hard way, and assume the quantity number in each math corpus as a measurement of difficulty level.
- RSM: Random-based framework for MWPs. We randomly select unlabeled samples and augment them with labeled samples.

Experimental results are demonstrated in Table 3. For space limitation, we only show labeled data with 25% and 50% ratio on Math23K dataset. It is clear that USM based model achieves the best performance than other baselines on the top of both DNS and GTS models. Interesting, we observe that random selection strategies obtains the lowest performance in all the scenarios. In our experiment, we also try other settings, e.g, increasing label data ratio to 75% and change batch size windows. However, we find the accuracy of RSM always lower than other strategies. We think the reason is that random selection strategies will incorporate some difficult samples, hence pseudo labels will serve as a noise, which further makes the logic reasoning process harder.

Compared with CSM and USM, it is clear that USM always performs better. As we claimed before, quantity length only partially demonstrates the complexity of math word problem, this straightforward measurement will can not reflect the authentic difficulty level for MWPs. This verifies our intuition that difficulty measurement is essential for semi-supervised MWPs task. Meanwhile, USM provides a quantitive assessment. The selected unlabeled data is high-quality and MWPs can benefit from data augmentation in this way.

Table 3: Effect of different data selection strategies on Math23K dataset.

Method	0.25		0.5	
	E-acc	A-acc	E-acc	A-acc
DNS-RSM	37.68	40.19	46.29	49.46
DNS-CSM	38.54	41.37	46.97	51.03
DNS-USM	44.69	48.91	52.14	56.90
GTS-RSM	45.81	52.74	53.64	62.08
GTS-CSM	46.92	54.85	54.97	64.02
GTS-USM	52.57	59.75	60.14	70.91

Table 4: Effects of each indirect supervision signals.

Method	0.25		0.5	
	E-acc	A-acc	E-acc	A-acc
DNS-USM	44.69	48.91	52.14	56.90
DNS-USM-w/o-QG	42.15	46.33	50.07	54.01
DNS-USM-w/o-GC	43.01	47.92	51.22	55.81
GTS-USM	52.57	59.75	60.14	70.91
GTS-USM-w/o-QG	50.04	57.07	58.92	68.78
GTS-USM-w/o-GC	51.76	58.90	59.23	69.00

3.2.2 Effects of Model Component.

In this section, we discuss the effectiveness of each indirect supervisions. Specifically, we perform the following methods on DNS-USM and GTS-USM models:

- USM-w/o-QG: USM without **Q**uestion **G**eneration loss.
- USM-w/o-GC: USM without **G**rammar **C**hecker loss.

Experimental results are shown in Table 4. We find that the proposed framework with both question generation loss and grammar checker loss outperforms best in the semi-supervised setting, on both the DNS and GTS model. We assume the reason to be that they are complementary to each other. Question Paraphrasing preserves semantic consistency information and make the equation logic reasoning understandable. Grammar checker loss enforces the format vadility of pseudo equations.

Besides, we observe that question paraphrasing contributes more than grammar checker. We assume there are three reasons: 1) The reward for grammar format checker is too sparse. The binary reward does not emphasize which pseudo equation

Table 5: Effects of each indirect supervision signals.

Context:	Car A and car B are going in opposite directions from cities C and D.Car A was travelling at an average speed of 75.5 km/h while car B was travelling at an average speed of 65.5 km/h. 4.5 hours later, the two cars met.
equation:	(75.5+65.5)*4.5
question:	what is the distance between the two cities?
paraphrasing	what is the distance between city c and d?
Context:	There are 5 people in Xiaofang's family. They consume 15.6 tons water and each tons cost 2 dollar.
equation:	15.6*2/5
question:	What is the average water charge each one?
paraphrasing	what is the bill each person?

token is invalid. 2) For GTS with a tree decoder, grammar checker loss only contributes a little since most of generated equations will follow a complete binary tree structure. It is obvious that grammar checker loss contributes more in DNS (seq2seq) model. 3) It is essential to enhance semantic meaning of each equation token. A similar observation is verified in (Zhang et al., 2020). From this study, we postulate that math problem in MWPs can provide some 'goal information' for equation generation. This verifies our intuition that equation generation has a high correspondence with problem description. This observation motivates an interesting direction on how to infer the logic equation from the relationship between both context and problem description.

3.3 Case Study

Here, we perform a case study on analyzing the generated questions in question paraphrasing stage. Table 5 represents the corresponding results sampled from the Math23k dataset. Each equation is the pseudo equation and model need to paraphrase the corresponding question. In the first example, the model learns three key words: 'distance', 'between', 'city c and city d'. It is obvious that the model not only reconstructs the initial question meanings but also emphasize some key words. In the second example, the paraphrasing questions is shorter than gold questions, however it learn substitution word for initial words, e.g., 'each person'.

4 Related Works

Math word problem, which combines knowledge understanding and logic reasoning, is a typical example in natural language understanding and has attracted researchers interests since 1960s (Bobrow, 1964). Due to labor-intensive involvements, earlier works mainly study on small datasets. These works are mainly grouped into statistical machine learning based (Koncel-Kedziorski et al., 2016; Hosseini et al., 2014) and semantic parsing based (Liguda and Pfeiffer, 2012) pipelines.

With the advance of deep learning, recent researchers utilized neural networks to solve this problem. One direction is to leverage the grammar structure of equations and ensure the grammar validity of equations. (Wang et al., 2017) was the first to incorporate vanilla seq2seq into MWPs, and its successive works designed a tree-based decoder either in an explicit (Xie and Sun, 2019; Liu et al., 2019) or implicit (Wang et al., 2019; Chiang and Chen, 2018) manner. Another direction is to enrich knowledge understanding from external source data. (Wu et al., 2020) proposed a knowledge-aware sequence-to-tree Network and the graphs are retrieved from external knowledge bases. However, since math equation is well-structured and lie with logical reasoning, these works are fully supervised. Recently, (Hong et al., 2020) proposed a learning-by-fixing framework which does not need math equation as supervisions, while gold-standard math answers are inevitable in their setting. Different from previous methods, our proposed method required no annotations (math equations or answers) for unlabeled data.

5 Conclusion

In this paper, we proposed an uncertainty aware semi-supervised framework on MWPs, which enables us to fully utilize unlabeled data. We leveraged model uncertainty to select reliable unlabeled data. Further, we introduced two additional indirect supervision signals to provide high-quality pseudo labeled samples. In addition, the experimental results on both Math23K and MAWPS dataset verified the effectiveness and generalization of our proposed framework. In the future, we will explore how to dynamically select reliable unlabeled data on MWPs. And it will be also interesting to design more informative reward signals to make the training stage more efficient.

References

José M Bernardo and Adrian FM Smith. 2009. *Bayesian theory*, volume 405. John Wiley & Sons.

Daniel G Bobrow. 1964. Natural language input for a computer problem solving system.

Ting-Rui Chiang and Yun-Nung Chen. 2018. Semantically-aligned equation generation for solving and reasoning math word problems. *arXiv preprint arXiv:1811.00720*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR.

Eduard Hofer, Martina Kloos, Bernard Krzykacz-Hausmann, Jörg Peschke, and Martin Woltereck. 2002. An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties. *Reliability Engineering & System Safety*, 77(3):229–238.

Yining Hong, Qing Li, Daniel Ciao, Siyuan Haung, and Song-Chun Zhu. 2020. Learning by fixing: Solving math word problems with weak supervision. *arXiv preprint arXiv:2012.10582*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference on Learning Representations*.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157.

Jierui Li, Lei Wang, Jipeng Zhang, Yan Wang, Bing Tian Dai, and Dongxiang Zhang. 2019. Modeling intra-relation in math word problems with different functional multi-head attentions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6162–6167.

Christian Liguda and Thies Pfeiffer. 2012. Modeling math word problems with augmented semantic networks. In *International Conference on Application of Natural Language to Information Systems*, pages 247–252. Springer.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. Tree-structured decoding for solving math word problems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2370–2379.

John P Miller and Wayne Seller. 1985. *Curriculum Perspectives and Practice.* ERIC.

Subhro Roy and Dan Roth. 2018. Mapping to declarative knowledge for word problem solving. *Transactions of the Association for Computational Linguistics*, 6:159–172.

Lei Shen and Yang Feng. 2020. Cdl: Curriculum dual learning for emotion-controllable response generation. *arXiv preprint arXiv:2005.00329*.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pages 387–395. PMLR.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to an expression tree. *arXiv preprint arXiv:1811.05632*.

Lei Wang, Dongxiang Zhang, Jipeng Zhang, Xing Xu, Lianli Gao, Bing Tian Dai, and Heng Tao Shen. 2019. Template-based math word problem solvers with recursive neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7144–7151.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854.

Qinzhuo Wu, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2020. A knowledge-aware sequence-to-tree network for math word problem solving. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7137–7146.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *International Joint Conference on Artificial Intelligence(IJCAI)*, pages 5299–5305.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2020. Graph-to-tree learning for solving math word problems. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F Wong, Yu Wan, and Lidia S Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944.

A Appendix

The pseudo algorithm of our USM framework.

Algorithm 1 The pseudo algorithm of our USM framework

Require: labeled data D_l, unlabeled data D_u.

Train a base model on label data D_l;
while not converge: **do**
    Apply trained base model on unlabeled data pools;
    Estimate uncertainty VAR(x_u) for unlabeled data;
    Calculate Var(D_u) via Eq.3 and β = mean(Var(D_u).
    if $\beta > 0.08$ **then**
        β = β - 0.01
    end if
    if VAR(x_u) $\leq \beta$ **then**
        Add current data into augmented pools.
    end if
    retrain the model on augmented data pools. Optimize labeled data with cross entropy loss and optimize selected data with Eq.8.
end while