MATHEMATICAL WORD PROBLEM GENERATION FROM COMMONSENSE KNOWLEDGE GRAPH AND EQUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

There is an increasing interest in the use of automatic mathematical word problem (MWP) generation in educational assessment. Different from standard natural question generation, MWP generation needs to maintain the underlying mathematical operations between quantities and variables, while at the same time ensuring the relevance between the output and the given topic. To address above problem we develop an end-to-end neural model to generate personalized and diverse MWPs in real-world scenarios from commonsense knowledge graph and equations. The proposed model (1) learns both representations from edge-enhanced Levi graphs of symbolic equations and commonsense knowledge; (2) automatically fuses equation and commonsense knowledge information via a self-planning module when generating the MWPs. Experiments on an educational gold-standard set and a large-scale generated MWP set show that our approach is superior on the MWP generation task, and it outperforms the state-of-the-art models in terms of both automatic evaluation metrics, i.e., BLEU-4, ROUGE-L, Self-BLEU, and human evaluation metrics, i.e, equation relevance, topic relevance, and language coherence.

1 INTRODUCTION

A mathematical word problem (MWP) is a coherent narrative that provides clues to the underlying correct mathematical equations and operations between variables and numerical quantities (Verschaffel et al., 2000; Cetintas et al., 2010; Moyer et al., 1984). MWPs challenge a student from a wide range of skills such as literacy skills for understanding the question, analytical skills for recognizing the problem type and applying arithmetical operators (Rembert et al., 2019; Moon-Rembert & Gilbert, 2019). Table 1 shows one such problem where students are asked to infer the counts of chickens and rabbits.

Table 1: An illustrative example of a MWP.

Mathematical Word Problem	Equations	Solutions
Chickens and rabbits were in the yard. Together they had 27 heads and 86 legs. How many chickens and rabbits were in the yard?[1]	x+y = 27 2x+4y = 86	x = 11 y = 16

In this paper, our objective is to automatically generate well-formed MWPs. Such automation will not only reduce the teachers' burden of manually designing MWPs, but provide students with a sufficiently large number of practice exercises, which help students avoid rote memorization (Williams, 2011; Wang & Su, 2016).

A large spectrum of models have been developed and successfully applied in broad area of natural question generation (NQG) (Pan et al., 2019; Li et al., 2018; Sun et al., 2018; Zhang & Bansal, 2019; Kurdi et al., 2020) and there has been a recent movement from the NQG community towards automatic generation of MWPs (Koncel-Kedziorski et al., 2016; Polozov et al., 2015; Zhou & Huang, 2019). For example, Koncel-Kedziorski et al. (2016) proposed a two-stage rewriting approach to edit existing human-authored MWPs. Polozov et al. (2015) conducted the MWP generation as a constrained synthesis of labeled logical graphs that represent abstract plots.

In general, there exist a large number of NQG models representing various text data and their syntax and semantics (Pan et al., 2019). However, automatic generation of MWPs still presents numerous challenges that come from special characteristics of real-world educational scenarios as follows:

- *Equation based symbolic representation*. MWP generation models need to not only generate fluent sentences but understand the mathematical variables, numerical quantities, operations, and their relations. Moreover, the models are supposed to be able to generalize to unsee equations.
- *Story plots in real-life scenarios*. Multiple studies have found that MWPs with real-life plots help conceptual knowledge understanding, discourse comprehension and children engagement (Carpenter et al., 1980; Jacque, 1996; Cummins et al., 1988; Davis-Dorsey et al., 1991; Polozov et al., 2015; Wang & Su, 2016; Rembert et al., 2019).
- *Narrative diversity*. Computerized educational assessment systems require diverse MWP results even given similar input equations, which helps prevent students from rote memorization(Deane & Sheehan, 2003).

To overcome the above challenges, in this paper, we present a novel neural generation model that aims to automatically generate coherent and diverse MWPs from given equations in students' real-life scenarios. More specifically, to fully understand the mathematical variables, numerical quantities, operations, and their relations, equations are transformed into an edge-enhanced Levi graph. We adopt the Gated Graph Neural Networks (GGNN) to learn representative embeddings from the equation based symbolic Levi graph. Meanwhile, the same procedure is applied on the external commonsense based knowledge graph (CSKG), which helps generate topic-relevant and semantically valid sentences in real-life settings. We choose to use the conditional Variational AutoEncoder (VAE) framework to generate MWPs from diversity promoting latent states. Furthermore, in the decoding stage, we develop a self-planning module to dynamically select and fuse information from both equations and commonsense knowledge, which improves syntax structure of generated MWP sentences.

Overall this paper makes the following contributions:

- We propose a GGNN based conditional VAE model for MWP generation. To the best of our knowledge, we are the first to introduce the combinational architecture of GGNN and condition VAE for MWP generation.
- We design a novel self-planning decoding module to wisely fuse information from equations and commonsense knowledge, which helps generate semantically and syntactically valid MWPs.
- The proposed model achieves state-of-the-art scores and outperforms existing methods by a significant margin on real-world educational MWP datasets from both automatic machinery and human evaluation metrics.

2 Related Work

2.1 Natural Question Generation

Previous research has directly approached the task of automatically generating questions for many useful applications such as augmenting data for the QA tasks (Tang et al., 2017; Zhao et al., 2018; Li et al., 2018; Sun et al., 2018; Zhang & Bansal, 2019), helping semantic parsing (Guo et al., 2018) and machine reading comprehension (Yu et al., 2020; Yuan et al., 2017), improving conversation quality (Mostafazadeh et al., 2016; Jain et al., 2018; Dong et al., 2019; Yao et al., 2012), and providing student exercises for education purposes (Deane & Sheehan, 2003; Koncel-Kedziorski et al., 2016; Zhou & Huang, 2019).

Various NQG methods are developed which can be divided into two categories: heuristic based approaches and neural network based approaches (Pan et al., 2019; Kurdi et al., 2020). The former generates questions in two stages: it first obtains intermediate symbolic representations and then constructs the natural language questions by either rearranging the surface form of the input sentence or generating with pre-defined question templates. The latter neural approaches view the NQG task

as a sequence-to-sequence (seq2seq) learning problem and jointly learn generation process in an end-to-end manner. Recently, advanced models are well studied to incorporate attention, copy, and coverage mechanisms into the NQG task (Du et al., 2017; Yao et al., 2018; Zhou et al., 2018).

2.2 Mathematical Word Problem Generation

Different from standard NQG tasks, generating MWPs not only needs the syntax, semantics and coherence of the output narratives, but requires understandings of the underlying symbolic representations and the arithmetic relationship between quantities. In general MWP generation approaches can be divided into three categories: (1) template based approaches; (2) rewriting based approaches; (3) neural network based approaches.

Template based approaches usually fall into a similar two-stage process: they first generalize an existing problem into a template or a skeleton, and then generate the MWP sentences from the templates (Deane & Sheehan, 2003; Williams, 2011; Wang & Su, 2016; Polozov et al., 2015; Bekele, 2020). Deane & Sheehan (2003) used semantic frames to capture both scene stereotypical expectations and semantic relationships among words and utilize a variant of second-order predicate logic to generate MWPs. Wang & Su (2016) leveraged the binary expression tree to represent the story of the MWP narrative and composed the natural language story recursively via a bottom-up tree traversal. Template based approaches heavily rely on the tedious and limited hand-crafted templates, leading to very similar generated results. This cannot meet the demand of a large number of high-quality and diverse MWPs.

Rewriting based approaches target the MWP generation problem by editing existing human-written MWP sentences to change their theme without changing the underlying story (Koncel-Kedziorski et al., 2016; Moon-Rembert & Gilbert, 2019). For example, Koncel-Kedziorski et al. (2016) proposed a rewriting algorithm to construct new texts by substituting thematically appropriate words and phrases. Rewriting based approaches are more flexible compared with templates based approaches. However, there are several drawbacks that prevent them from providing the large number of personalized MWPs. First, the generation process is based on existing MWPs, which significantly limits the generation ability. Second, students easily fall into rote memorization since it is too trivial to notice that the underlying mathematical equations are still unchanged.

Recent attempts have been focused on exploiting neural network based approaches that generating MWPs from equations and topics in an end-to-end manner (Zhou & Huang, 2019; Liyanage & Ranathunga, 2019; 2020). Zhou & Huang (2019) designed a neural network with two encoders to fuse information of both equations and topics and dual-attention mechanism to generate relevant MWPs. Liyanage & Ranathunga (2020) tackled the generation problem by using the long short term memory network with enhanced input features, such as character embeddings, word embeddings and part-of-speech tag embeddings.

The closest work to our approach is Zhou & Huang (2019) and the main differences are as follows: (1) Zhou & Huang (2019) directly encode the equation by a single-layer bidirectional gated recurrent unit (GRU), while we first convert equations into Levi graph and conduct the encoding by the GGNN model; (2) instead of directly using the pre-trained embeddings of similar words given the topic, we choose to learn the topic relevant representations from an external CSKG; and (3) we choose to use the VAE framework to promoting more diverse results.

3 Learning from Commonsense Knowledge and Equations

Our objective is to automatically generate a significant number of diverse MWPs in students' real-life scenarios from valid equations. Similar to Polozov et al. (2015), we support the personalized generation in which students (or teachers) can determine the story plots of MWPs by specifying topics. A topic indicates a type of real-world scenarios, such as animals, sports, etc.

Following most work on NQG tasks, we adopt the encoder-decoder architecture, shown in Figure 1. The input includes a set of equations and a knowledge graph with a specific topic. We construct Levi graphs (Levi, 1942) from symbolic equations and the CSKG respectively (See Section 3.1). After that, we employ GGNNs to extract the full graph structure information about equations and real-life story plots (See Section 3.2). Then, we generate target sentence by a conditional VAE with a self-

planning module (See Section 3.3). The self-planning module enables the decoder to pay different portions of attention to the equations and the CSKG.

Please note that in this paper, we focus on generating MWPs with linear equations of two variables without any constraint. Our framework can be easily generalized into MWPs with different numbers of variables with little modification.

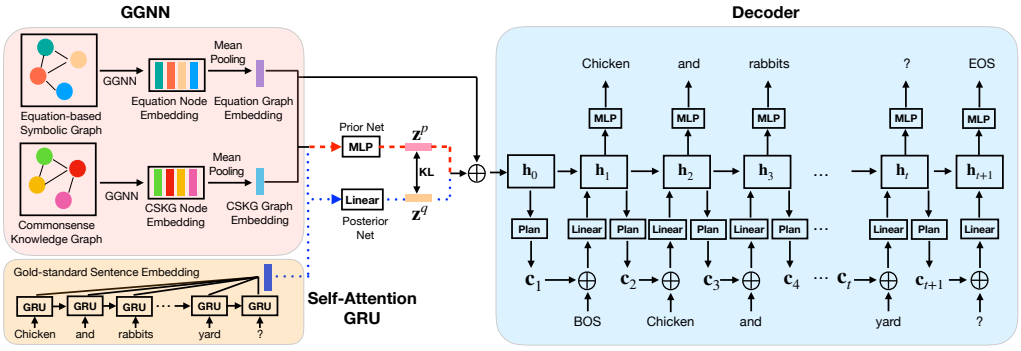

Figure 1: The overview of the proposed framework. The blue dot line (or red dash line) is only enabled in the training (or inference) stage. $\oplus$ denotes the vector concatenation. **Linear** represents the linear transformation and **Plan** denotes the self-planning module (discussed in Section 3.3).

3.1 Levi Graph Construction

3.1.1 Equation Based Symbolic Graph

The equation based symbolic graph is designed to capture the relations among mathematical variables and numerical quantities, and build connections between mathematical variables and the corresponding commonsense knowledge. In this work, we consider the linear equations (with two variables) behind the MWPs as $ax + by = m; cx + dy = n$, where x and y are the variables and a, b, c, d, m, and n are positive integer quantities. Equation variants are discussed in Section 4.1.

Equations are first converted to a symbolic graph as shown in Figure 2 (a). In the symbolic graph, edge labels, i.e., *Add*, *Mul*, etc. representing the mathematical relations play important roles in the MWP generation. In order to well capture such relations, we model the edge labels as explicit nodes. Following previous work in Beck et al. (2018), we transform the symbolic graph into its equivalent edge-enhanced Levi graph (Levi, 1942) by adding two nodes for each labeled edge. One node denotes the forward direction of the relation and one represents the reverse. By adding reverse nodes, we encourage more information flow from the reverse direction, in the same way RNN-based encoders benefit from right-to-left propagation. Furthermore, we explicitly add self-loop edges to each node in the Levi graph. The symbolic Levi graph is depicted in Figure 2 (b).

3.1.2 Commonsense Based Knowledge Graph

In order to generate plots in students' life scenarios and provide the personalized flexibility, we utilize implicit knowledge from an external CSKG. With the help of CSKG, students or teachers are able to set their own preferences when generating MWPs. Such preferences are referred to as topics, such as zoo, transportation, school life, etc. Moreover, the CSKG improves the generation quality by alleviating ill-informed wordings or sentences. For instance, in spite of no grammatical errors, it makes no sense to have "rabbits are in the ocean". Figure 2 (c) illustrates a sample of a CSKG with a topic of *poultry*. Similar to the Levi graph construction procedure in Section 3.1.1, we introduce additional nodes for relations in CSKG and add reverse and self-loop edges. The CSKG Levi graph is shown in Figure 2 (d).

3.2 Gated Graph Neural Networks Encoding

Following the success of GGNN models (Li et al., 2015; Beck et al., 2018; Ruiz et al., 2019), we use GGNNs to capture both the mathematical relations among variables and quantities and the real-life associations among entities in the MWPs. Specifically, let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be an edge-enhanced Levi

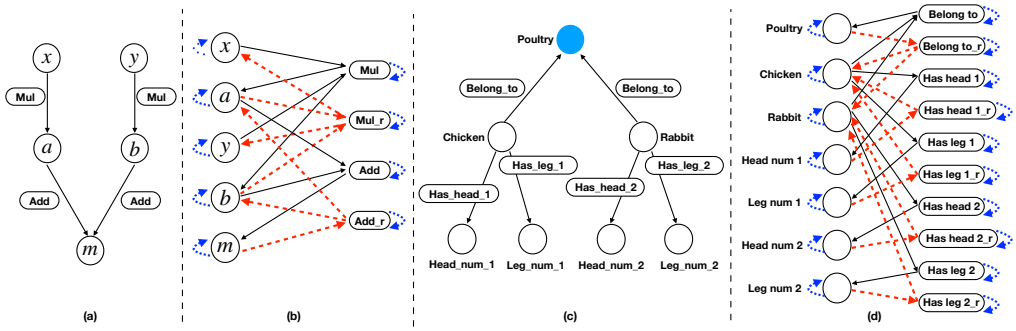

Figure 2: (a) a sample symbolic graph of equation "ax+by = m"; (b) the edge-enhanced Levi graph of the same equation; (c) an illustrative sample of the CSKG under topic *poultry*; (d) the corresponding edge-enhanced Levi graph of CSKG. The red dash arrows represent the reverse edges and the blue dot arrows represent the self-loop edges. *Mul* and *Add* denote the multiply and addition operations. The subscript "_r" denotes the artificially added reverse nodes.The blue node in Figure 2(c) denotes the given topic.

graph where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges. Let $a^{v,u}$ be the similarity between node v and node u from its row-wise normalized adjacent matrix. Given an input Levi graph $\mathcal{G}$ that may represent either the equations or the CSKG, the basic recurrence of the GGNN model is defined as follows:

$$\mathbf{g}_0^v = \mathbf{e}_0^v; \quad \boldsymbol{\gamma}_t^v = \sum_{u \in N_{(v)}} a^{v,u} \mathbf{g}_{t-1}^u; \quad \mathbf{z}_t^v = \sigma(\mathbf{W}^z \boldsymbol{\gamma}_t^v + \mathbf{U}^z \mathbf{g}_{t-1}^v); \quad \mathbf{r}_t^v = \sigma(\mathbf{W}^r \boldsymbol{\gamma}_t^v + \mathbf{U}^r \mathbf{g}_{t-1}^v)$$

$$\widetilde{\mathbf{g}_t^v} = \tanh(\mathbf{W}^h \boldsymbol{\gamma}_t^v + \mathbf{U}^h (\mathbf{r}_t^v \odot \mathbf{g}_{t-1}^v)); \quad \mathbf{g}_t^v = (1 - \mathbf{z}_t^v) \odot \mathbf{g}_{t-1}^v + \mathbf{z}_t^v \odot \widetilde{\mathbf{g}_t^v}$$

where $\mathbf{e}_0^v$ denotes the initial embedding of node v. $N_{(v)}$ is the set of neighbor nodes for v and σ is the sigmoid function. $\odot$ is the component-wise multiplication function and $\mathbf{z}_t^v$ and $\mathbf{r}_t^v$ are gating vectors.

Let $\mathbf{G}_0 = [\mathbf{g}_0^1; \mathbf{g}_0^2; \cdots; \mathbf{g}_0^{|\mathcal{V}|}]$ be the initial word embedding matrix of all the nodes and $\mathbf{G}_n$ be the matrix of representation of node embeddings from the above GGNN model after n iterations, i.e., $\mathbf{G}_n = [\mathbf{g}_n^1; \mathbf{g}_n^2; \cdots, \mathbf{g}_n^{|\mathcal{V}|}]$. Similar to He et al. (2016), we ease the downstream learning tasks with embedding augmentation. We apply a linear transformation on the concatenation of $\mathbf{G}_0$ and $\mathbf{G}_n$, i.e., $\mathbf{G}_* = \mathbf{W}_*[\mathbf{G}_0; \mathbf{G}_n]$. Such augmented node representations contain abstract context information, which are used in our language generator in Section 3.3. Let $\mathbf{G}_*^e$ and $\mathbf{G}_*^k$ be the augmented GGNN embeddings of the equations and the CSKG. Meanwhile we apply a mean pooling operation over $\mathbf{G}_*^e$ and $\mathbf{G}_*^k$ to get the graph-level equation representation ($\mathbf{g}_*^e$) and CSKG representation ($\mathbf{g}_*^k$).

3.3 Conditional Variational AutoEncoder with Self-Planning Module

In this section, we introduce our VAE architecture with the self-planning module for the MWP generation. Our self-planning module makes dynamic fusion on the learned representations of equations and CSKG to generate the MWPs.

Let Y be the random variable representing the texts of MWPs and Z be the diversity promoting latent variable of the distribution of the MWPs. Let C be the random variable representing the conditions of both the explicit equations and the implicit CSKG learned from GGNNs. We model the MWP generation by the conditional distribution as follows: $p(Y|C) = \int p(Y|C,Z)p(Z|C)dZ$ where $p(Y|C,Z)$ is the MWP generator and $p(Z|C)$ is the prior net. Since the integration of Z is intractable, we apply variational inference and optimize the evidence lower bound as follows:

$$\log p(Y|C) \geq \mathbb{E}_{q(Z|C,Y)}\Big[\log p(Y|C,Z)\Big] - D_{KL}\Big(q(Z|C,Y)||p(Z|C)\Big) \tag{1}$$

where $D_{KL}(\cdot||\cdot)$ denotes the KL-divergence.

Following conventions, we assume both the prior net and posterior net of Z following the isotropic Gaussian distributions, i.e., $p(Z|C) \sim \mathcal{N}(\boldsymbol{\mu}^p, \sigma^p \mathbf{I})$ and $q(Z|C,Y) \sim \mathcal{N}(\boldsymbol{\mu}^q, \sigma^q \mathbf{I})$. The prior net

only encodes the given conditions of both the explicit equations and the implicit CSKG while the posterior net encodes both given conditions and the texts of MWPs. Both the prior net and the posterior net are built upon the GGNNs shown in Figure 1 as follows:

$$[\boldsymbol{\mu}^p; \log \boldsymbol{\sigma}^p] = \mathbf{MLP}\big([\mathbf{g}^e_*; \mathbf{g}^k_*]\big); \quad [\boldsymbol{\mu}^q; \log \boldsymbol{\sigma}^q] = \mathbf{W}^q\big([\mathbf{g}^e_*; \mathbf{g}^k_*; \mathbf{GRU}(\mathbf{y})]\big) + \mathbf{b}^q$$

Since there may exist more than one expression logic which cover the same input but in different order, we capture such diversity of reasonable presentations with both latent variable Z and input graphs C. Different samples of Z will lead to different self-planning results. In this work, we realize the self-planning mechanism in our attention-based GRU decoder to capture different portions of input information from the equations and the CSKG. To start the decoding process, we initialize the hidden state ($\mathbf{h}_0$) as $\mathbf{h}_0 = [\mathbf{z}; \mathbf{g}^e_*; \mathbf{g}^k_*]$ where $\mathbf{z}$ is sampled from the posterior net $q(Z|C, Y) \sim \mathcal{N}(\boldsymbol{\mu}^q, \boldsymbol{\sigma}^q\mathbf{I})$ and the prior net $p(Z|C) \sim \mathcal{N}(\boldsymbol{\mu}^p, \boldsymbol{\sigma}^p\mathbf{I})$ during the training and inference procedures respectively.

At each decoding time step t, we use the attention mechanism to conduct the self-planning between explicit symbolic equations and implicit CSKG. The self-planning module takes the decoder's current hidden state ($\mathbf{h}_t$), node representations of equations ($\mathbf{G}^e_*$) and CSKG ($\mathbf{G}^k_*$) as input and outputs the context-aware planning state ($\mathbf{c}_t$) of the current time step. Specifically, we compute $\mathbf{c}_t$ as follows:

$$\mathbf{c}_t = \beta_t * \mathbf{c}^e_t + (1 - \beta_t) * \mathbf{c}^k_t; \quad \beta_t = softmax(\mathbf{h}_t); \quad \mathbf{c}^e_t = \sum_{v \in \mathcal{V}^e} \alpha^e_{t,v} \mathbf{g}^e_v; \quad \mathbf{c}^k_t = \sum_{v \in \mathcal{V}^k} \alpha^k_{t,v} \mathbf{g}^k_v$$

$$\alpha^e_{t,v} = \exp(\mathbf{o}^e_{t,v}) / \sum_{v' \in \mathcal{V}^e} \exp(\mathbf{o}^e_{t,v'}); \quad \alpha^k_{t,v} = \exp(\mathbf{o}^k_{t,v}) / \sum_{v' \in \mathcal{V}^k} \exp(\mathbf{o}^k_{t,v'})$$

$$\mathbf{o}^e_{t,v} = {\mathbf{v}^e}^\top \tanh(\mathbf{W}^e \mathbf{h}_t + \mathbf{U}^e \mathbf{g}^e_v); \quad \mathbf{o}^k_{t,v} = {\mathbf{v}^k}^\top \tanh(\mathbf{W}^k \mathbf{h}_t + \mathbf{U}^k \mathbf{g}^k_v)$$

where β_t represents the self-planning distribution at time step t.

The final context vector is the fusion of the symbolic and commonsense knowledge graphs. The next-step hidden state ($\mathbf{h}_{t+1}$) is the combination of current hidden state ($\mathbf{h}_t$), self-planning context state ($\mathbf{c}_t$) and the representation of currently generated word ($\mathbf{w}_t$), i.e, $\mathbf{h}_{t+1} = \mathbf{GRU}(\mathbf{h}_t, \mathbf{W}^d[\mathbf{c}_t; \mathbf{w}_t] + \mathbf{b}^d)$ where $\mathbf{W}^d$ and $\mathbf{b}^d$ are the linear transformation matrix and the bias term. We further generate the next word by feeding hidden state $\mathbf{h}_{t+1}$ to linear transformation and softmax layer to get the next-token probability distribution.

Our model is trained end-to-end by optimizing eq.(1). It consists (1) maximizing the probability of ground-truth sequence texts, which promotes the predictions generated by the posterior net and the MWP generator closer to the distribution of the gold-standard data; and (2) minimizing the KL-divergence between posterior distribution ($p(Z|C, Y)$) and prior distribution ($p(Z|C)$).

4 EXPERIMENTS

In this work, we crawled 1,275 MWPs of linear equations from a third-party website. It covers 47 topics and the average length of a MWP is 48 words. We randomly select 196 of them as our gold-standard test (GT) set. We make our code publicly available at `https://tinyurl.com/y3teywts`.

We use following evaluation metrics: (1) *BLEU-4*: the 4-gram overlap score against gold-standard sentences (Papineni et al., 2002); (2) *ROUGE-L*: the overlap of longest common subsequence between candidate and gold-standard sentences (Lin, 2004); (3) *Self-BLEU*: the diversity measurement of averaging BLEU scores of four generated MWP pairs given the same input (Zhu et al., 2018).

Meanwhile, we conduct two human evaluation studies to comprehensively evaluate the quality of the generated MWPs. First, we ask evaluators to rate from the following aspects ranging from 1 to 3 (Chen et al., 2019; Wang & Wan, 2019): (1) *Equation Relevance*: how relevant between the MWP and the input equations? (2) *Topic Relevance*: how relevant between the MWP and the given topic? and (3) *Language Coherence*: whether the MWP is coherent and well-organized. We use the average scores from three human evaluators as our final results.

During training, we use linear KL annealing technique following Fu et al. (2019) to alleviate the KL collapse problem and apply scheduled sampling to alleviate the exposure bias problem in GRU training (Bengio et al., 2015). Implementation details are listed in Appendices A.1 - A.2.

Our approach of *Ma*thematical word problem generation from commonsense *K*nowledge and *E*quations is referred to as "*MaKE*". We compare against the following baselines: (1) the template based method, i.e., *Template*; (2) conditional VAE that captures the diversity in the encoder and uses latent variables to learn a distribution over potential intents, i.e., *CVAE* (Zhao et al., 2017); (3) the state-of-the-art pre-trained language model with a shared Transformer network and self-attention masks, i.e., *UniLM* (Dong et al., 2019); and (4) a standard Transformer-based seq2seq model, i.e., *Transformer* (Vaswani et al., 2017). More details are provided in Appendix A.3.

4.1 Results and Analysis

Evaluation Results on GT Set. Results on the GT set are listed in Table 2, which shows that our *MaKe* outperforms all other methods in terms of both automatic and human evaluation metrics. Specifically, from Table 2, we find: (1) comparing *MaKE* and *Template*, *Template* doesn't perform well in language coherence and topic relevance. This is because the MWP templates are stereotyped. Mismatches between the template context and the re-filled words lead to incoherent texts. (2) with rich representations of equations and CSKG, *MaKE* is able to better capture the mathematical relations and improve MWP quality with real-life plots under the given topic.

Turing Test Results on GT Set. For each existing MWP in the GT set, we generate a new MWP of the same equations but with a different topic. We show such pairs to the human evaluators and ask them to distinguish which one is the generated MWP. We measure the results of this artificial "Turing Test" via *Fool Ratio*, i.e., the fraction of instances in which a model is capable of fooling the evaluators. Ideally, perfect MWP generation will lead to random guesses and the ideal *Fool Ratio* would be 50%. Finally, we get an averaged *Fool Ratio* of 38.93% (39.8%, 41.3% and 35.7% from three annotators respectively). This demonstrates that the generation quality is 77.86% (38.93/50) as good as the quality from human teachers.

Ablation Study

Table 2 shows the results of ablation study. Without the self-planning module, the performance of our model drops by 2.1% in *BLEU-4* and 3.3% in *ROUGE-L*, which indicates its effectiveness. These scores also drop when CSKG is removed, which indicates that the representations of CSKG not only improve the coherence but help form valid MWPs in real-life scenarios. The *MaKE w/o CSKG* approach achieves the best *Self-BLEU* score but the worst human evaluation scores, which means that none of the diverse sentences are valid enough for MWPs.

Table 2: Evaluation results on GT set and ablation study. *Rel.* and *Coh.* are short for relevance and coherence.

Method	BLEU-4	ROUGE-L	Self-BLEU	Equation Rel.	Topic Rel.	Language Coh.
Template	33.1	50.8	68.2	2.881	2.864	2.694
CVAE	30.6	48.6	68.6	2.781	2.821	2.532
UniLM	26.0	44.4	76.3	2.325	2.646	2.051
Transformer	30.9	49.9	77.9	2.859	2.869	2.721
MaKE w/o planning	31.4	48.0	73.5	2.821	2.878	2.709
MaKE w/o CSKG	20.3	40.5	**62.9**	2.721	2.241	2.190
MaKE	**33.5**	**51.3**	68.1	**2.886**	**2.912**	**2.743**

Qualitative Case Study

Because of the GGNN encodings of equations, our *MaKE* model is able to handle a wide range of mathematical relations, including both addition and subtraction, i.e., a, b, m, c, d and n may be either positive or negative in $ax + by = m; cx + dy = n$. We quantitatively compare the generation quality of *MaKE* with other baselines and the results are shown in Table 3. Furthermore, we show the diverse generation results of *MaKE* qualitatively in Table 4. Additional examples can be found in Appendices A.4 - A.5. As we can see, (1) *CVAE* and *Transformer* cannot interpret the equations correctly and fail to generate desired MWPs; (2) our *MaKE* approach is able to generate diverse enough MWPs in real-life scenarios.

Table 3: Illustrative examples of the MWP generation comparison with unsee equations. () represents the question that the student needs to solve. There is no results from *Template* because it doesn't work on unseen equations. The incorrect part is highlighted in red color.

Equations: x-y=6; 2x-4y=35;	**Topic**: Rowing boat
CVAE	The water park has 6 more small boats and 6 more big boats, for a total of 35 people. If there are 2 more people in a big boat and 4 more in a small boat, and 35 more people in a small boat than in a big boat, then there are () big boats and () small boats.
UniLM	Two teachers lead 35 participants number of participants is 35?
Transformer	There are 6 boats in the water park, each small boat can accommodate 2 people and each big boat can accommodate 4 people. Six boats can accommodate total of 35 people. There are () big boats and () small boats.
MaKE	Teacher Mr.Huang and his 35 students come to row the boat. They find 6 more small boats than big ones. There are 35 more students in the small boat than in the big boat. How many big boats are there?

Table 4: An illustrative example of the diverse MWP generation made by *MaKE*. () represents the question that the student needs to solve.

Equations: x=y; 2x+4y=48; **Topic**: Poultry
1. Rabbits and chicken are in one cage. The number of rabbits is 0 less than that of chickens, they have 48 legs in total, how many rabbits and chickens in cage?
2. Rabbit and chicken, we know that the number of two kinds of animals are the same, the total number of legs is 48, so how many rabbits and chickens are in the farm?
3. Chicken and rabbits are in the same cage, chicken heads are 0 more than rabbit heads, and there are 48 legs. May I ask there are () chickens and () rabbits?
4. Monkey King showed the magic to the monkeys. A group of chickens and a group of rabbits emerged. After counting, they found that there were 48 legs. If we know that the number of chickens is 0 more than that of rabbits. Then how many chickens and rabbits for each?

Human Evaluation Results on Large-scale Generated MWPs. Besides evaluations on the GT set, which is usually limited in educational scenarios (Xu et al., 2019), we conduct evaluations on the large-scale generated results.

We randomly create 100 valid linear equations and ensure that none of them appears in our training set. Meanwhile, we select top 30 common real-life topics. For each pair of equation and topic, we generate 5 MWPs accordingly and therefore, we obtain 15,000 MWPs. We conduct a human evaluation to assess the quality of these generated MWPs and the results are shown in Table 5. We can see that our method achieves the best results and outperforms baseline models with a large margin.

Table 5: Results on the large-scale generated MWP data. There is no results from *Template* because it doesn't work on unseen equations.

Method	Equation Rel.	Topic Rel.	Language Coh.
CVAE	1.478	2.444	1.878
UniLM	1.200	1.894	1.394
Transformer	1.983	2.544	2.589
MaKE	**2.289**	**2.654**	**2.672**

5 Conclusion

In this paper, we presented a neural encoding-decoding architecture for MWP generation. Comparing with the existing NQG algorithms, the advantages of our *MaKE* are: (1) it extracts intrinsic representations of both the equation based symbolic graph and the commonsense based knowledge graph; (2) it automatically selects and incorporates informations from equations and knowledge graphs during the decoding process; and (3) it is able to generate relevant, coherent and personalized MWPs in students' real-life scenarios. Experimental results on real-world educational MWP data sets demonstrated that *MaKE* outperforms other state-of-the-art NQG approaches in terms of both automatic evaluation metrics and human evaluation metrics. In the future, we plan to explore the MWP generation problems for more mathematical variables with high-order operations.

REFERENCES

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 273–283, 2018.

Andinet Assefa Bekele. Automatic generation of amharic math word problem and equation. *Journal of Computer and Communications*, 8(8):59–77, 2020.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 1171–1179, 2015.

Thomas P Carpenter, Mary Kay Corbitt, Henry S Kepner, Mary Montgomery Lindquist, and Robert E Reys. Solving verbal problems: Results and implications from national assessment. *The arithmetic teacher*, 28(1):8–12, 1980.

Suleyman Cetintas, Luo Si, Yan Ping Xin, Dake Zhang, Joo Young Park, and Ron Tzur. A joint probabilistic classification model of relevant and irrelevant sentences in mathematical word problems. *Journal of Educational Data Mining*, 2(1):83–101, 2010.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. Reinforcement learning based graph-to-sequence model for natural question generation. In *International Conference on Learning Representations*, 2019.

Denise Dellarosa Cummins, Walter Kintsch, Kurt Reusser, and Rhonda Weimer. The role of understanding in solving word problems. *Cognitive psychology*, 20(4):405–438, 1988.

Judy Davis-Dorsey, Steven M Ross, and Gary R Morrison. The role of rewording and context personalization in the solving of mathematical word problems. *Journal of Educational Psychology*, 83(1):61, 1991.

Paul Deane and Kathleen Sheehan. Automatic item generation via frame semantics: Natural language generation of math word problems. 2003.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pp. 13063–13075, 2019.

Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*, 2017.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

Daya Guo, Yibo Sun, Duyu Tang, Nan Duan, Jian Yin, Hong Chi, James Cao, Peng Chen, and Ming Zhou. Question generation from sql queries improves neural semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1597–1607, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ensign Jacque. Linking life experiences to classroom math. 1996.

Unnat Jain, Svetlana Lazebnik, and Alexander G Schwing. Two can play this game: visual dialog with discriminative question generation and answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5754–5763, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. A theme-rewriting approach for generating algebra word problems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1617–1628, 2016.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204, 2020.

Friedrich Wilhelm Levi. *Finite geometrical systems: six public lectues delivered in February, 1940, at the University of Calcutta*. University of Calcutta, 1942.

Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. Visual question generation as dual task of visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6116–6124, 2018.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Vijini Liyanage and Surangika Ranathunga. A multi-language platform for generating algebraic mathematical word problems. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pp. 332–337. IEEE, 2019.

Vijini Liyanage and Surangika Ranathunga. Multi-lingual mathematical word problem generation using long short term memory networks with enhanced input features. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 4709–4716, 2020.

DeKita G Moon-Rembert and Juan E Gilbert. Illmatics: A web-based math word problem generator for students' distal and proximal interests. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pp. 842–848. Association for the Advancement of Computing in Education (AACE), 2019.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. Generating natural questions about an image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1802–1813, 2016.

John C Moyer, Larry Sowder, Judith Threadgill-Sowder, and Margaret B Moyer. Story problem formats: Drawn versus verbal versus telegraphic. *Journal for Research in Mathematics Education*, pp. 342–351, 1984.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. Recent advances in neural question generation. *arXiv preprint arXiv:1905.08949*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Oleksandr Polozov, Eleanor O'Rourke, Adam M Smith, Luke Zettlemoyer, Sumit Gulwani, and Zoran Popović. Personalized mathematical word problem generation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

DeKita Moon Rembert, Naja A Mack, and Juan E Gilbert. Exploring the needs and interests of fifth graders for personalized math word problem generation. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, pp. 592–597, 2019.

Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Gated graph convolutional recurrent neural networks. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE, 2019.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University, 1999.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3930–3939, 2018.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Lieven Verschaffel, Brian Greer, and Erik De Corte. *Making sense of word problems*. Swets & Zeitlinger Lisse, 2000.

Ke Wang and Zhendong Su. Dimensionally guided synthesis of mathematical word problems. In *IJCAI*, pp. 2661–2668, 2016.

Tianming Wang and Xiaojun Wan. T-cvae: transformer-based conditioned variational autoencoder for story completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 5233–5239. AAAI Press, 2019.

Sandra Williams. Generating mathematical word problems. In *2011 AAAI Fall symposium series*, 2011.

Guowei Xu, Wenbiao Ding, Jiliang Tang, Songfan Yang, Gale Yan Huang, and Zitao Liu. Learning effective embeddings from crowdsourced labels: An educational case study. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1922–1927. IEEE, 2019.

Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and Yanjun Wu. Teaching machines to ask questions. In *IJCAI*, pp. 4546–4552, 2018.

Xuchen Yao, Emma Tosch, Grace Chen, Elnaz Nouri, Ron Artstein, Anton Leuski, Kenji Sagae, and David Traum. Creating conversational characters using question generation tools. *Dialogue & Discourse*, 3(2):125–146, 2012.

Jianxing Yu, Xiaojun Quan, Qinliang Su, and Jian Yin. Generating multi-hop reasoning questions to improve machine reading comprehension. In *Proceedings of The Web Conference 2020*, pp. 281–291, 2020.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 15–25, 2017.

Shiyue Zhang and Mohit Bansal. Addressing semantic drift in question generation for semi-supervised question answering. 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)*, pp. 2495–2509, 2019.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 654–664, 2017.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3901–3910, 2018.

Qingyu Zhou and Danqing Huang. Towards generating math word problems from equations and topics. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 494–503, 2019.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Sequential copying networks. *arXiv preprint arXiv:1807.02301*, 2018.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1097–1100, 2018.