A Mathematical Word Problem Generator with Structure Planning and Knowledge Enhancement

Longhu Qin
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
qlonghu@mail.ustc.edu.cn

Jiayu Liu
School of Data Science, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
jy251198@mail.ustc.edu.cn

Zhenya Huang*
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
huangzhy@ustc.edu.cn

Kai Zhang
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
sa517494@mail.ustc.edu.cn

Qi Liu
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
qiliuql@ustc.edu.cn

Binbin Jin
Huawei Cloud Computing Technologies Co., Ltd.
Hangzhou, Zhejiang, China
jinbinbin1@huawei.com

Enhong Chen
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
cheneh@ustc.edu.cn

ABSTRACT

Automatically generating controllable and diverse mathematical word problems (MWPs) which conform to equations and topics is a crucial task in information retrieval and natural language generation. Recent deep learning models mainly focus on improving the problem readability but overlook the mathematical logic coherence, which tends to generate unsolvable problems. In this paper, we draw inspiration from the human problem-designing process and propose a **Ma**thematical structure **P**lanning and **K**nowledge enhanced **G**eneration model (**MaPKG**), following the "plan-then-generate" steps. Specifically, we propose a novel dynamic planning module to make sentence-level equation plans and a dual-attention mechanism for word-level generation, incorporating equation structure representation and external commonsense knowledge. Extensive experiments on two MWP datasets show our model can guarantee more solvable, high-quality, and diverse problems. Our code is available at https://github.com/KenelmQLH/MaPKG.git

CCS CONCEPTS

• **Computing methodologies → Natural language generation**.

KEYWORDS

MWP generation, planning mechanism, knowledge enhancement

ACM Reference Format:
Longhu Qin, Jiayu Liu, Zhenya Huang, Kai Zhang, Qi Liu, Binbin Jin, and Enhong Chen. 2023. A Mathematical Word Problem Generator with Structure Planning and Knowledge Enhancement. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3591937

1 INTRODUCTION

Automatic problem generation has attracted much attention in information retrieval and natural language generation fields, which could provide important educational resources for several applications [9, 15, 16]. In this paper, we study the task of automatically generating mathematical word problems (MWPs), which not only asks for semantic understanding [6, 14, 25] of the specific equations and topics, but requires mathematical logic to generate controllable

*Corresponding author.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591937

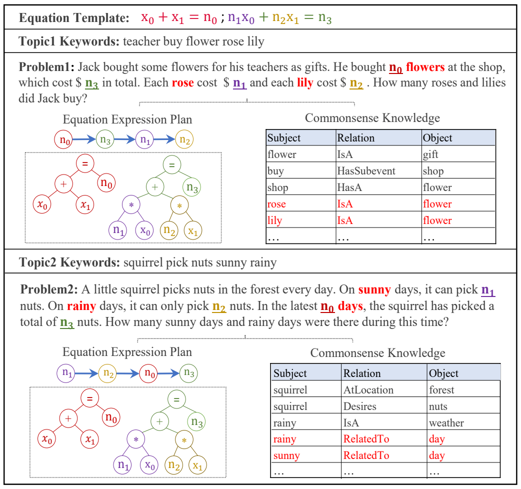

Figure 1: Examples of diverse MWP generation.

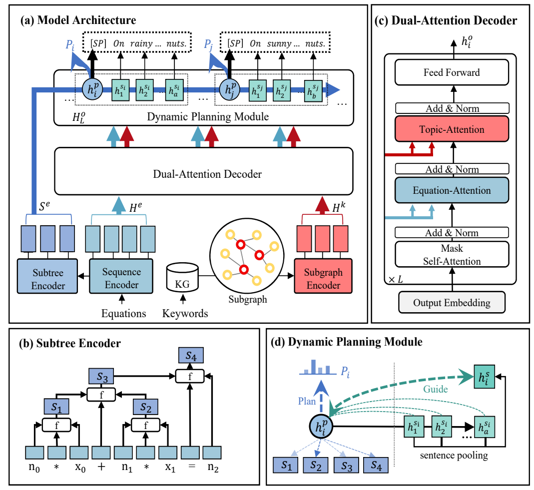

Figure 2: Framework of MaPKG.

and diverse problems. As shown in Figure 1, given the same equation templates but different topic keywords, we can create problems that describe mathematical information in diverse scenarios.

Toward this goal, there are several efforts in the task including rule-based methods and neural network-based ones. Specifically, the earlier rule-based methods [8, 18, 23] always generate a problem with predefined math rules or text templates, and however, they generally suffer from manual construction cost and limited template diversity. Recently, researchers change the attention to neural network approaches, which follow the sequence-to-sequence architecture to generate diverse MWPs [17, 28]. Moreover, some works explore the possibilities including the pre-trained language model [24], retrieval-based generation [2], and commonsense enhancement[3] etc. Although they have achieved great success, they generally focuses on improving the problem readability. Their generation process may overlook the mathematical logic coherent among MWPs, and tend to generate problems that may be unsolvable in practice.

In this paper, we draw inspiration from the problem-designing process of human educators. On one hand, human experts always follow the "plan-then-generate" principle [11, 19] in real-world situations. Specifically, before writing down a problem, they usually make an explicit plan to express equations in a logical order. For Problem1 in Figure 1, the equation "$x_0 + x_1 = n_0$" (marked red) is selected first and then described as the sentence "He bought n_0 flowers ...", followed by the green part to be generated. On the other hand, when generating a specific sentence, it is essential to associate commonsense knowledge, which not only helps select proper keywords but also enrich the description. In Figure 1, if we know that both "rose" and "lily" are similar concepts related to "flower", they could be more likely to describe the variables "x_0" and "x_1" respectively after describing "n_0" as the number of flowers.

However, it is non-trivial for machines to carry out this human-style process for MWP generation. First, the token-level equation sequences fail to accurately reflect their mathematical logic. In Figure 1, the reasonable plan for Problem1 follows the subtree-level order marked as "red-green-purple-yellow" steps, rather than the token-level order. Second, introducing more topic keywords may lead to a harder knowledge application process, since we should not only comprehend the concepts [26] based on commonsense knowledge but also combine math information for description.

To this end, we propose a novel **Ma**thematical structure **P**lanning and **K**nowledge enhanced **G**eneration model (**MaPKG**) for MWP generation following the "plan-then-generate" process with the encoder-decoder architecture. Specifically, in the encoder, we introduce the subtree structure and external knowledge to represent the equations and keywords respectively. In the decoder, we first propose a novel dynamic planning module to make sentence-level expression plans based on equation subtrees. Then, we design a dual-attention mechanism to fuse equation information and topic knowledge to generate problem word by word. Extensive experiments on two MWP datasets verify that our MaPKG improves the generation results in terms of solvability, quality and diversity.

2 METHOD

2.1 Task Definition

Given mathematical equation templates $\boldsymbol{E} = \{e_1, ..., e_n\}$ and topic keywords $\boldsymbol{T} = \{w_1, ..., w_m\}$, the MWP generator aims to generate a descriptive problem $\boldsymbol{y} = \{y_1, ..., y_l\}$ by:

$$\boldsymbol{y} = \text{argmax}_{\hat{y}} P(\hat{y} \mid \boldsymbol{E}, \boldsymbol{T}). \quad (1)$$

Two requirements should be met in this process. First, $\boldsymbol{y}$ can be solved by the input equations $\boldsymbol{E}$. Second, $\boldsymbol{y}$ is described as coherent narrative text related to the input topic $\boldsymbol{T}$.

2.2 Model Architecture

Figure 2 shows the framework of MaPKG, which mainly consists of two multi-grained equation encoders, a keyword subgraph encoder, a dynamic planning module, and a dual-attention decoder.

2.2.1 Equation Representation. We consider that equations contain two kinds of information. First, an equation can be directly viewed as a sequence of numbers, variables, and operators. Second, the operators determine the relationship between numbers and variables, which form structural subtrees [7] that imply specific sentence-level descriptions (e.g., the subtree "$x_0 * n_1$" are grounding for sentence "Each rose cost n_1"). To this end, we represent each equation $e \in \boldsymbol{E}$ from two aspects, namely sequential token representations and hierarchical subtree representations.

Table 1: The statistics of the datasets.

Statistics	Lmwp-G	Hmwp-G
Num. problems	5447	5491
Avg Num. words	39.6	60.1
Num. templates	48	2144
Avg Num. equations	2.0	1.3
Avg Num. keywords	8.84	8.80
Avg Num. Concepts	64.6	49.66
Avg Num. Triples	98.24	71.13

For sequential token representations, we use a Bi-GRU based encoder to represent token sequence embeddings as $\boldsymbol{H}^e = \{h^e_1, ..., h^e_{|e|}\}$.

For subtree representations, we first convert equation e into a binary expression tree and denote each subtree i as a triplet $T_i = (o, l, r)$. Then we propose a subtree encoder to learn the subtree embeddings in a bottom-up way as shown in Figure 2 (b). Specifically, the subtree embedding s_i of i is obtained by:

$$s_i = W_s[h^e_o; s_l; s_r] + b_s, \tag{2}$$

where $h^e_o \in \boldsymbol{H}^e$ is the representation of operator o, s_l, s_r are embeddings of the left child l and right child r respectively. For every leaf node, we set $s_* = h^e_* \in \boldsymbol{H}^e$. The hierarchical subtree embeddings are donated as $\boldsymbol{S}^e = \{s_1, ..., s_b\}$, where b is the number of subtrees.

2.2.2 Knowledge-aware Keyword Representation. Keywords provide essential semantic information to generate a problem. However, it is still not enough to only perceive them isolatedly, which might lead to improper expression of math information (e.g., generate that the number of "roses" is the sum of "flowers" and "lilies").

Therefore, when representing the keywords, we retrieve external commonsense knowledge to promote the understanding of them. Specifically, given keywords $\boldsymbol{T}$, we first regard them as central concepts and extract their K-hop neighbor concepts $\boldsymbol{V}$ from the public knowledge bases ConceptNet [1] and HowNet[2], which form a keyword subgraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \boldsymbol{T} \cup \boldsymbol{V}$ and $\mathcal{E}$ are edges taken from the knowledge bases. Then, we learn the node representations $\{g^v \mid v \in \mathcal{V}\}$ with a keyword subgraph encoder, which is implemented with GGNN [12]. After N iterations of information passing on $\mathcal{G}$, we concat the initial and N-th iteration node presentations g^v_0, g^v_N for each node v, i.e. $h^k_v = W_k[g^v_0; g^v_N] + b_k$, and obtain the knowledge-aware keyword representations as $\boldsymbol{H}^k = \{h^k_v \mid v \in \boldsymbol{T}\}$.

2.2.3 Dynamic Planning Module. To accurately express the logic of equations in problems, we generate subtree plans as a skeleton to guide the sentence expression order. Specifically, inspired by [4, 5, 13], we treat the plans as additional tokens denoted as $[SP]$ and generate them with words in problem y dynamically by a dual-attention mechanism described in Section 2.2.4, which derives a hidden state h^p for each $[SP]$ as its latent representation.

In this module, we aim to ensure that (1) h^p indeed conveys the plan information about one or more subtrees in equations $\boldsymbol{E}$ and (2) h^p guides the generation of the current sentence (e.g., "Jack bought ... n_0 flowers ..." is grounded on the current plan "$x_0 + x_1 = n_0$").

For the first goal, we introduce a prediction task to determine which subtree(s) the plan h^p indicates. Specifically, we calculate the probability that h^p relates to the subtree T_i by a pointer-network:

$$P(T_i) = \sigma(W_1^T \tanh(W_2[s_i; h^p] + b_1) + b_2), \tag{3}$$

[1] https://conceptnet.io
[2] https://openhownet.thunlp.org

which induces a subtree planning loss as ($\widetilde{g}$ is the golden plans):

$$\mathcal{L}_{plan} = -\textstyle\sum_{T_i \in \widetilde{g}} \log P(T_i) - \sum_{T_i \notin \widetilde{g}} \log(1 - P(T_i)). \tag{4}$$

For the second goal, we design a guidance loss to enhance the dependence of the sentence on the current plan. Specifically, we use the mean squared error to minimize the latent plan representation h^p and the sentence representation h^s. h^s is obtained by average pooling the hidden states of tokens in the generated sentence.

$$\mathcal{L}_{sent} = MSE(h^p, h^s). \tag{5}$$

With the above proper planning and forced guiding, the decoder below can guarantee the equation logic in word generation, to ensure the problem quality and solvability.

2.2.4 Dual-Attention Decoder. Our decoder adopts the basic Transformer manner to conduct word-level generation. Specifically, we propose a novel dual-attention mechanism in it to combine the equation information with the knowledge-aware keyword information. As shown in Figure 2(c), we design an equation attention layer and a topic attention layer as follows:

$$H'_l = H_l + MultiHead_E(H_l, \boldsymbol{H}^e, \boldsymbol{H}^e), \tag{6}$$

$$H''_l = H'_l + MultiHead_T(H'_l, \boldsymbol{H}^k, \boldsymbol{H}^k), \tag{7}$$

where H_l represents the hidden states from the mask self-attention layer. With L decoder blocks, the dual-attention mechanism can fuse the equation information and topic information iteratively. Finally, the hidden states in the last decoder layer h^o are passed to another feed-forward layer with softmax to estimate output distribution and generate the problem words. Especially, if $[SP]$ is generated, h^o is taken as the latent plan representation h^p described in Section 2.2.3. The generation loss is defined as follows:

$$\mathcal{L}_{LM} = -\textstyle\sum_{t=1}^{l} \log P(y_t \mid y_{<t}, \boldsymbol{E}, \boldsymbol{T}), \tag{8}$$

Finally, we jointly minimize the following loss with hyperparameters α, β to balance sentence planning and word generation:

$$\mathcal{L} = \mathcal{L}_{LM} + \alpha\mathcal{L}_{plan} + \beta\mathcal{L}_{sent}. \tag{9}$$

3 EXPERIMENTS

3.1 Experimental Dataset and Setup

3.1.1 Datasets. We conduct our experiments based on two MWP datasets. (1) Lmwp [17] is a dataset with two linear equations and two unknown variables for each problem. (2) Hmwp [20] consists of hybrid MWPs including both one-known and two-unknown, which contains more various equation templates and longer problems. Based on them, we extract topic keywords from each problem with jionlp[3] and annotate the subtree positions in equations as the golden plans. Table 1 summarizes the basic statistics of the annotated datasets Lmwp-G and Hmwp-G.

3.1.2 Experimental Setup. We set the embedding dim as 512 and the number of transformer layers as 6. The keyword sub-graphs are constructed by 1-hop neighbors (i.e., $K = 1$) , and $N = 2$ in GGNN. Hyperparameter α and β are both set to 0.5.

3.1.3 Baseline and Evaluation. We compare our model against several strong baselines: (1) **CVAE** [27] is a GRU-based sequence-to-sequence model with Conditional VAE. (2) **S2S-GRU** [1] is a GRU-based sequence-to-sequence model with Attention. (3) **MAG-NET** [28] is a MWP generator with entity-enforced loss. (4) **S2S-TF** [22] is a standard Transformer-based sequence-to-sequence model.

[3] http://www.jionlp.com/

Table 2: Automatic evaluation on MWP generation.

Dataset	Lmwp-G					Hmwp-G				
Metric	Solvability		Quality		Diversity	Solvability		Quality		Diversity
	Equ-Acc↑	Ans-Acc↑	BLEU↑	METEOR↑	Self-BLEU↓	Equ-Acc↑	Ans-Acc↑	BLEU↑	METEOR↑	Self-BLEU↓
CVAE	0.5350	0.6341	25.40	0.4908	0.7203	0.1750	0.2250	29.61	0.4843	0.6705
S2S-GRU	0.6310	0.7169	27.40	0.5015	0.7278	0.2370	0.2910	31.77	0.5004	0.6493
MAGNET	0.6305	0.7004	25.06	0.4839	0.6832	**0.3404**	0.3891	25.37	0.4314	0.6534
S2S-TF	0.7000	0.7800	29.04	0.5397	0.7133	0.3030	0.3720	39.13	0.5796	0.6630
BART	0.6213	0.7426	**31.64**	**0.5595**	0.6791	0.3050	0.3830	41.93	0.6050	**0.6033**
MaPKG	**0.7440**	**0.8490**	30.79	0.5410	**0.6780**	0.3320	**0.4082**	**42.34**	**0.6149**	0.6378
w/o SP	0.7366	0.8379	30.54	0.5395	0.6850	0.3184	0.3965	42.05	0.6072	0.6442
w/o KG	0.7403	0.8434	29.94	0.5273	0.6869	0.3320	0.3984	40.61	0.5975	0.6474
w/o DA	0.7016	0.8158	29.60	0.5266	0.7002	0.2891	0.3652	38.85	0.5800	0.6561

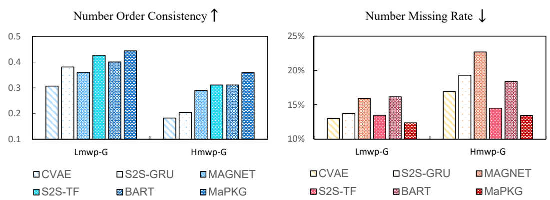

Figure 3: Analysis for numbers in generated MWPs.

(5) **BART** [10] is one of the most popular generative pre-trained language models, and we choose BART-base for comparison.

We conduct the automatic evaluation in three aspects: solvability, language diversity, and quality. For solvability, we use equation accuracy (**Equ-Acc**) and answer accuracy (**Ans-Acc**) to measure whether the generated MWPs can be solved by the input equations. The equation accuracy is checked using a SOTA MWP solver [21], where the predicted equations are compared with the original equations. The answer accuracy is computed by the answers of unknowns in equations. For language diversity, we use **Self-BLEU** [29] to measure the diversity of generated problems. For language quality, we select **BLEU**(average of BLEU-1,2,3,4) and **METEOR**.

3.2 Results and Analysis

3.2.1 Main Results. Table 2 reports the evaluation results and we observe that MaPKG outperforms the baselines on most occasions. Specifically, for problem solvability, MaPKG achieves the best performance overall, which verifies the effectiveness of the "plan-then-generate" principle for generating logically reasonable MWPs. For language quality and diversity, MaPKG and BART achieve the best results, proving that our MaPKG is competitive with the pre-training language models in language by knowledge enhancement.

We also conduct ablation studies in Table 2. Specifically, we introduce "w/o SP" which omits the subtree-based dynamic planning module, "w/o KG" which replaces the keyword subgraph encoder with the initial keyword embeddings , and "w/o DA" which replaces the dual-attention decoder with a standard Transformer decoder.

We conclude the results as follows. First, all components contribute to MWP generation since removing any module leads to performance degradation. Second, "w/o DA" diminishes all metrics significantly, implying that the fusion of equation and keyword is the basis for correctly describing math information. Third, "w/o SP" diminishes the results greatly in equation consistency. It indicates that our proposed planning module is necessary and crucial for generating solvable problems. Fourth, the performance of "w/o KG" shows that knowledge enhancement benefits language diversity.

Table 3: Examples of generated MWPs.

Equation: $x_0 + x_1 = n_0; n_1 * x_0 - n_2 * x_1 = n_3$
Keywords: Grandma green red cake food cost
S2S-TF: In the past, cakes are rare treats. Each green bean cake cost n_1 yuan and each red bean cake cost n_2 yuan. A family bought n_0 cakes at the cost of n_3 yuan. How many green bean cakes and red bean cakes did they buy? (✗)
BART: Grandma Wang spent n_0 yuan to buy some cakes. Each green bean cake cost n_1 yuan and each red bean cake cost n_2 yuan. There were n_3 more green bean cakes than red bean cakes. How many red bean cakes and red bean cakes did Wang buy? (✗)
MaPKG: Grandma Li went to the street to buy n_0 pieces of cake. Each green bean cake cost n_1 yuan, each red bean cake cost n_2 yuan. The green bean cakes cost more n_3 yuan than the red bean cakes. How many red bean cakes? (✓)

3.2.2 Analysis of Planning. MaPKG's planning effectiveness is demonstrated by analyzing the numbers in generated MWPs. We assess the logical order of equations of generated MWPs by measuring number order consistency with labeled MWPs. The number missing rate is also computed to determine the omission of numbers in generated MWPs. Our results in Figure 3 indicate that MaPKG outperforms other models in both metrics, indicating that it not only produces logical plans but also encourages number expression.

3.2.3 Case Study. A representative example in Table 3 shows that S2S-TF and BART generate problems with good language expression but unsatisfying equation consistency. S2S-TF misrepresents the operator "−" as "the sum of cost", and BART misinterprets "n_0" as the "total price of cakes". Conversely, MaPKG precisely perceives equation subtrees with the planning module and incorporates commonsense knowledge to avoid these situations.

4 CONCLUSION

In this paper, we proposed a novel MWP generation model (MaPKG) following the "plan-then-generate" steps. Specifically, we introduced the subtree structure and external knowledge into representation modeling. Then, we proposed a dynamic planning module to make sentence-level expression plans based on equation subtrees. Next, we designed a dual attention mechanism to fuse equations and topic knowledge in word-level generation. Extensive experiments on two MWP datasets verified that our MaPKG improved the solvability, quality and diversity of generated problems.

ACKNOWLEDGMENTS

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. 62106244, No. U20A20229), and the University Synergy Innovation Program of Anhui Province (No. GXXT-2022-042).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Tianyang Cao, Shuang Zeng, Xiaodan Xu, Mairgup Mansur, and Baobao Chang. 2022. DISK: Domain-constrained Instance Sketch for Math Word Problem Generation. *arXiv preprint arXiv:2204.04686* (2022).

[3] Tianyang Cao, Shuang Zeng, Songge Zhao, Mairgup Mansur, and Baobao Chang. 2021. Generating math word problems from equations with topic consistency maintaining and commonsense enforcement. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III 30*. Springer, 66–79.

[4] Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long Text Generation by Modeling Sentence-Level and Discourse-Level Coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 6379–6393. https://doi.org/10.18653/v1/2021.acl-long.499

[5] Zhe Hu, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Hua Wu, and Lifu Huang. 2022. PLANET: Dynamic Content Planning in Autoregressive Transformers for Long-form Text Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2288–2305. https://doi.org/10.18653/v1/2022.acl-long.163

[6] Zhenya Huang, Xin Lin, Hao Wang, Qi Liu, Enhong Chen, Jianhui Ma, Yu Su, and Wei Tong. 2021. Disenqnet: Disentangled representation learning for educational questions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 696–704.

[7] Zhenya Huang, Qi Liu, Weibo Gao, Jinze Wu, Yu Yin, Hao Wang, and Enhong Chen. 2020. Neural mathematical solver with enhanced formula structure. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1729–1732.

[8] Nabila Ahmed Khodeir, Hanan Elazhary, and Nayer Wanas. 2018. Generating story problems via controlled parameters in a web-based intelligent tutoring system. *The International Journal of Information and Learning Technology* 35, 3 (2018), 199–216.

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

[10] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[11] Junyi Li, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Knowledge-based review generation by coherence enhanced text planning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 183–192.

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

[13] Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021. Conversations Are Not Flat: Modeling the Dynamic Information Flow across Dialogue Utterances. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 128–138. https://doi.org/10.18653/v1/2021.acl-long.11

[14] Xin Lin, Zhenya Huang, Hongke Zhao, Enhong Chen, Qi Liu, Hao Wang, and Shijin Wang. 2021. Hms: A hierarchical solver with dependency-enhanced understanding for math word problem. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4232–4240.

[15] Jiayu Liu, Zhenya Huang, Xin Lin, Qi Liu, Jianhui Ma, and Enhong Chen. 2022. A Cognitive Solver with Autonomously Knowledge Learning for Reasoning Mathematical Answers. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 269–278.

[16] Jiayu Liu, Zhenya Huang, Chengxiang Zhai, and Qi Liu. 2023. Learning by Applying: A General Framework for Mathematical Reasoning via Enhancing Explicit Knowledge Learning. *arXiv preprint arXiv:2302.05717* (2023).

[17] Tianqiao Liu, Qiang Fang, Wenbiao Ding, Hang Li, Zhongqin Wu, and Zitao Liu. 2021. Mathematical Word Problem Generation from Commonsense Knowledge Graph and Equations *(Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing)*. Association for Computational Linguistics, 4225–4240. https://doi.org/10.18653/v1/2021.emnlp-main.348

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

[19] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 6908–6915.

[20] Jinghui Qin, Lihui Lin, Xiaodan Liang, Rumin Zhang, and Liang Lin. 2020. Semantically-Aligned Universal Tree-Structured Solver for Math Word Problems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 3780–3789. https://doi.org/10.18653/v1/2020.emnlp-main.309

[21] Yibin Shen, Qianying Liu, Zhuoyuan Mao, Zhen Wan, Fei Cheng, and Sadao Kurohashi. 2022. Seeking Diverse Reasoning Logic: Controlled Equation Expression Generation for Solving Math Word Problems. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Online only, 254–260. https://aclanthology.org/2022.aacl-short.32

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[23] Ke Wang and Zhendong Su. 2016. Dimensionally Guided Synthesis of Mathematical Word Problems.. In *IJCAI*. 2661–2668.

[24] Zichao Wang, Andrew Lan, and Richard Baraniuk. 2021. Math Word Problem Generation with Mathematical Consistency and Problem Context Constraints *(Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing)*. Association for Computational Linguistics, 5986–5999. https://doi.org/10.18653/v1/2021.emnlp-main.484

[25] Kai Zhang, Qi Liu, Hao Qian, Biao Xiang, Qing Cui, Jun Zhou, and Enhong Chen. 2023. EATN: An Efficient Adaptive Transfer Network for Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (Jan 2023), 377–389. https://doi.org/10.1109/TKDE.2021.3075238

[26] Kun Zhang, Guangyi Lv, Le Wu, Enhong Chen, Qi Liu, and Meng Wang. 2021. Ladra-net: Locally aware dynamic reread attention net for sentence semantic matching. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[27] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. 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)*. Association for Computational Linguistics, Vancouver, Canada, 654–664. https://doi.org/10.18653/v1/P17-1061

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

[29] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 1097–1100.