Recall and Learn: A Memory-augmented Solver for Math Word Problems

Shifeng Huang *
CVTE Research
huangshifeng@cvte.com

Jiawei Wang *†
CVTE Research
wangjiawei0531@gmail.com

Jiao Xu
CVTE Research
xujiao@cvte.com

Da Cao
Hunan University
caoda0721@gmail.com

Ming Yang
CVTE Research
yangming@cvte.com

Abstract

In this article, we tackle the math word problem, namely, automatically answering a mathematical problem according to its textual description. Although recent methods have demonstrated their promising results, most of these methods are based on template-based generation scheme which results in limited generalization capability. To this end, we propose a novel human-like analogical learning method in a recall and learn manner. Our proposed framework is composed of modules of memory, representation, analogy, and reasoning, which are designed to make a new exercise by referring to the exercises learned in the past. Specifically, given a math word problem, the model first retrieves similar questions by a memory module and then encodes the unsolved problem and each retrieved question using a representation module. Moreover, to solve the problem in a way of analogy, an analogy module and a reasoning module with a copy mechanism are proposed to model the interrelationship between the problem and each retrieved question. Extensive experiments on two well-known datasets show the superiority of our proposed algorithm as compared to other state-of-the-art competitors from both overall performance comparison and micro-scope studies.

1 Introduction

The task of Math Word Problem (MWP) aims at automatically solving a mathematical question according to its textual description. Given a problem description, a model needs to understand the relevant quantities and reason the corresponding expression, which is a difficult task because it requires the model to learn mathematics knowledge from the labeled problem and generalize the knowledge to the unseen problems.

In fact, great efforts have been made to address the MWPs in the research community. Boosted by the proliferation of deep learning techniques, Seq2Seq-based models have been developed to solve MWPs. Wang et al. (2017) presented a large-scale MWP dataset Math23K and proposed an RNN-based framework with a number mapping technique, which aims to generate a math template first, and then fill the extracted number from the problem into the slots of the generated template to obtain an expression. This two-stage method is widely used as a baseline by the latest papers, such as Math-EN (Wang et al., 2018), GTS (Xie and Sun, 2019), Graph2Tree (Zhang et al., 2020b), Ape (Zhao et al., 2020) and so on (Wang et al., 2019b; Li et al., 2019).

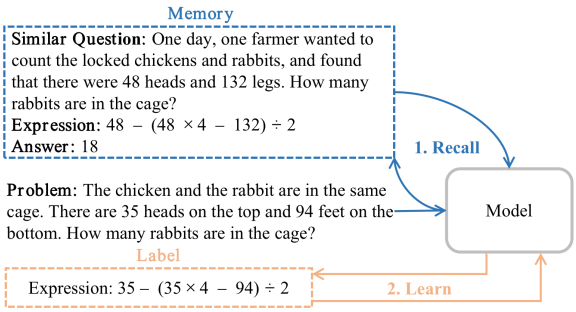

Figure 1: Illustration of our proposed framework for solving math word problems in a recall and learn manner.

Despite its value and significance, the math word problem has not been well addressed due to the following challenges: 1) Although promising results have been reported, the aforementioned models all use the template-based framework to solve MWPs, such a two-stage process may introduce systematic cumulative errors. In light of this, how to solve MWPs properly without using the template is a non-trivial task. 2) Furthermore, instead of learning through a single training example, the way human learn often rely on the so-called analogical learning method, which is able to explore the inherent laws between various cases and generalize them to new examples (Schwartz et al., 2016; Hope et al., 2017). Therefore, how to combine the analogical learning

*Both authors contributed equally to this research.
†Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 786–796
November 7–11, 2021.

method in a unified framework is worth exploring.

To address the aforementioned issues, as revealed in Figure 1, we design a novel memory-augmented model named REAL (short for "REcall And Learn") to solve the MWP task in an end-to-end manner. REAL is able to recall some familiar questions that have been solved when solving a new problem, and learns to generate a similar solution in an analogical way. Specifically, REAL model first initializes a memory module by a dataset formed with questions and their expressions. When solving a problem, the memory module is utilized to retrieve the most similar questions as references according to the unsolved problem. Next, a representation module is proposed to extract item memories of the unsolved problem and the retrieved question. Thereafter, we employ an analogy module to construct relational memory based on the item memories. Finally, a reasoning module is applied to generate the expression of the unsolved problem by combining the generation and copy mechanisms. Extensive experiments show that we have achieved competitive performance on MWP task. Moreover, our proposed model is able to improve the performance by retrieving more questions, which shows the model has the ability to learn by analogy.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, this is the first model that learns to solve math word problems using human-like analogical learning way.
- We develop a novel memory-augmented framework combined with the copy mechanism, REAL, to solve MWPs in a recall and learn manner, in which the model is composed of modules of memory, representation, analogy and reasoning.
- Extensive experiments are conducted on two well-known datasets, and the results showed that the REAL model not only achieves competitive performance on MWP task, but also demonstrates the unique ability of learning by analogy. Meanwhile, we have released the code to facilitate the research community.[1]

2 Related Work

In this section, we briefly review some literatures that are tightly related to our work, namely, math word problems and memory-augmented generative methods.

2.1 Math Word Problems

In the MWP task, the algorithms are designed to calculate a mathematical expression based on the textual description of mathematical problems. Therefore, the methods of natural language processing can be widely used in MWP task. Most of existing models adopt an encoder-decoder framework, where the encoder is designed as a bidirectional RNN and the decoder is designed as a unidirectional RNN. For example, Wang et al. (2017) constructed a large dataset and proposed a Seq2Seq model that shows the superiority over previous works. Wang et al. (2018) proposed an equation normalization technique to solve the order-duplicated problem and bracket-duplicated problem. Wang et al. (2019b) designed a tree-structure model to predict the suffix expression of MWPs, which reduces the target space of the problem. Xie and Sun (2019) proposed a tree-structured gated recurrent unit as decoder, which passes the information through the expression tree in both top-down and bottom-up manners. Zhang et al. (2020b) proposed a graph encoder to enrich the quantity representations in the problem, and decode the expression by a tree structure decoder. Zhao et al. (2020) presented a new large-scale and template-rich MWP dataset Ape210K and proposed a strong Seq2Seq model, which achieves state-of-the-art performance on both the Math23K and Ape210K datasets. However, these models highly rely on a method that extracting numbers from the question, and then mapping numbers to the slots of the generated templates. Such a two-stage process will introduce some systematic errors to the model.

Therefore, we consider exploring the pipeline of generating expression directly instead of utilizing the template as an intermediate process, in which the model may gain more information from the question description and benefit from the end-to-end training strategy.

2.2 Memory-augmented Generative Methods

In the text generation task, there are mainly two types of models, one is based on retrieval (Zhou et al., 2016, 2018; Zhang et al., 2018; Chen et al., 2019b; Wang et al., 2019a), and the other is based on generation (Qian et al., 2018; Zhou and Wang, 2018; Dong et al., 2019; Han et al., 2019). The retrieval algorithm can solve a particular task by

[1] https://github.com/sfeng-m/REAL4MWP

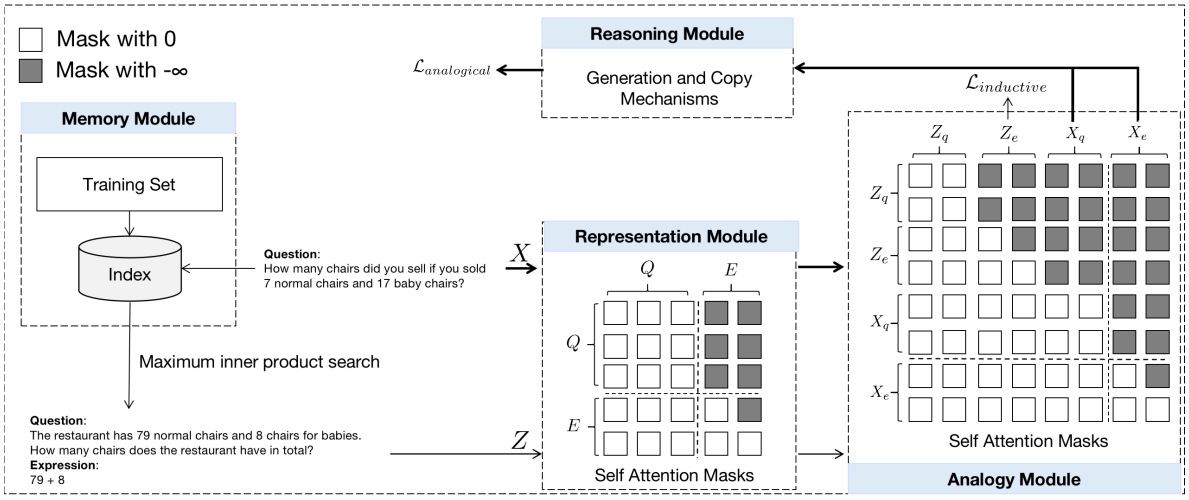

Figure 2: The illustration of our proposed REAL framework, which is composed of modules of memory, representation, analogy and reasoning. For an unsolved problem X, we first use maximum inner product search to find a similar question Z from the memory module. And then the solution is generated with the copy mechanism in an analogical manner.

constructing a knowledge base, which has high scalability. However, the retrieval-based approach cannot generate arbitrary results, which restricts the generation space of the model. In addition, the generative framework is able to store the knowledge in the model with the form of parameters, which has a certain generalization ability. However, in the knowledge-intensive task, it is difficult to remember all the knowledge in the parameters of generative model. To this end, many researchers attempted to combine retrieval and generation methods for text generation task (Zhang et al., 2017; Zhu et al., 2019; Lewis et al., 2020; Koncel-Kedziorski et al., 2019; Chen et al., 2019a; Zhou et al., 2020). In particular, Zhang et al. (2017) proposed a memory-augmented neural model for Chinese poetry generation, which investigates the contribution of memory. Zhu et al. (2019) have demonstrated a retrieval-enhanced response generation approach for a dialogue system, which makes use of informative content in retrieved results to generate new responses. Lewis et al. (2020) proposed a retrieval-augmented generation method where the parametric memory is a pre-trained Seq2Seq model and the non-parametric memory is a pre-trained neural retriever.

Our work is inspired by the success of incorporating memory into the generative model, showing memory-augmented model is capable of achieving strong performance in MWPs. Moreover, with the help of the memory module, our proposed model is able to solve the MWPs by analogy, which opens up a new research direction on MWP task.

3 Method

The framework of REAL is presented in Figure 2. In general, our proposed framework is composed of four key components: 1) Memory Module is constructed with a pre-trained model and is able to return top-K similar questions given a math word problem. 2) Representation Module is used to represent each token of the problem and the question in an inductive manner. 3) Analogy Module is utilized to aggregate the information of the problem and the retrieved question for better generating the correct expression. 4) Reasoning Module is combined with a copy mechanism that acts as a decoder to generate each token of the expression based on the input sequence.

3.1 Problem Formulation

We denote a problem as $X = \{X_q, X_e\}$, where the subscript q and e indicate the question description and mathematical expression respectively. X_q is a sequence of word tokens $X_q = \{x_q^1, x_q^2, \cdots, x_q^L\}$, where L is the length of the question description. We let its K retrieved similar questions $\mathcal{Z} = \{Z^1, Z^2, \cdots, Z^K\}$ where $Z^i = \{Z_q^i, Z_e^i\}$. For each unsolved problem, the goal is to predict the token of X_e at each time step t, namely $y_t \in \mathcal{V} \cup X_q$, where $\mathcal{V}$ is a generated vocabulary.

3.2 Memory Module

Aiming at solving a math word problem based on its similar retrieved questions, we employ a memory module to acquire external knowledge for enhancing the learning ability of the unsolved prob-

lem. The memory module is a non-parameter retriever, which is defined as the following formulation:

$$p(Z|X) \approx p(Z_q|X_q) = \frac{e^{f(X_q)^T f(Z_q)}}{\sum_{Z_q} e^{f(X_q)^T f(Z_q)}}, \quad (1)$$

where $f(\cdot)$ is a Word2Vec (Mikolov et al., 2013) model followed by a mean pooling technique that can represent a question description as a dense vector. In order to retrieve the similar question Z_q given an unsolved problem X_q, we first normalize each vector and perform the MIPS (maximum inner product search) algorithm, which is implemented similar to the FAISS library (Johnson et al., 2017). Note that we utilize $p(Z_q|X_q)$ to approximate $p(Z|X)$ because only the problem description is provided in the testing stage.

3.3 Representation Module

The representation module is leveraged to summarize the representation of the problem and each retrieved question, which is called item memory. The module is constructed by the Transformer (Vaswani et al., 2017) block with a casual mask that similar to the settings of UniLM (Dong et al., 2019), which can learn a bidirectional encoder and a unidirectional decoder simultaneously. Specifically, we perform a causal masking mechanism to allow each position in the expression to attend to previous positions, which preserve the auto-regressive property during decoding. In addition, we realize the representation of each token by summing the token, segment and position embeddings, which is similar to the approach of BERT model (Devlin et al., 2019). Next, follows the settings of UniLM (Dong et al., 2019), to avoid the information-leakage problem during training, we use causal masks to ensure that the representation of each token in expression is only related to the previous states, as shown in Figure 2.

Therefore, in the training stage, given a problem $\{X_q, X_e\}$ with its corresponding retrieved questions $\{Z_q, Z_e\}$, the representation module is employed to acquire the item memories $\mathbf{X}_q$, $\mathbf{X}_e$, $\mathbf{Z}_q$ and $\mathbf{Z}_e$ with the same dimension of 768 respectively, which efficiently learns the representations of the problem and each retrieved question in an inductive manner.

3.4 Analogy Module

In order to achieve the way of analogical learning, the model needs to aggregate contextual information from the item memories of the unsolved problem and the retrieved questions. Therefore, we first concatenate item memories to form input features $\{\mathbf{Z}_q, \mathbf{Z}_e, \mathbf{X}_q, \mathbf{X}_e\}$ and preprocess the input features using the mechanisms of position encoding and segment encoding. Thereafter, based on the length of the input sequences Z_q, Z_e, X_q and X_e, a casual mask can be constructed similar to the approach of representation module, as shown in Figure 2. The purpose of the casual mask is to enhance the analogical learning capability by focusing the attention of unsolved problem on the retrieved questions. In addition, the expression part in a casual mask is designed to only attend to the previous token, which avoids the information-leakage problem. Lastly, we utilize a Transformer network that similar to the representation module for learning relational memories by analogy. Thereinto, the output states of the analogy module are denoted as relational memories $\hat{\mathbf{Z}}_q$, $\hat{\mathbf{Z}}_e$, $\hat{\mathbf{X}}_q$ and $\hat{\mathbf{X}}_e$ respectively. Note that $\hat{\mathbf{Z}}_q$ and $\hat{\mathbf{Z}}_e$ are the outputs of the last layer, and the $\hat{\mathbf{X}}_q$ and $\hat{\mathbf{X}}_e$ are the outputs of penultimate layers.

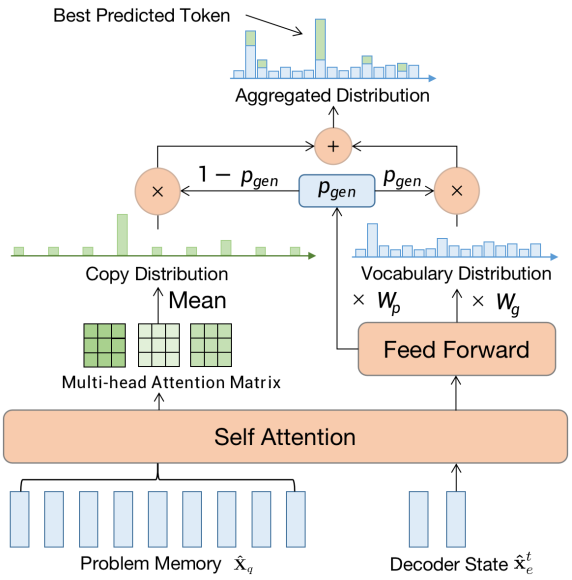

Figure 3: The overview of the reasoning module with a copy mechanism.

In order to extract the knowledge from the retrieved question $\{Z_q, Z_e\}$, we further employ a classifier $C \in \mathbb{R}^{768\times|V|}$ to solve the question description of Z_q and propose an auxiliary loss $\mathcal{L}_{inductive}$ to navigate the learning direction of the analogy module, which is formulated as follows:

$$\mathcal{L}_{inductive} = -\sum_{t}^{N} log p_{\theta_a}(z_e^t | Z_q, z_e^{1:t-1}), \quad (2)$$

where z_e^t indicates the t^{th} token of Z_e, and θ_a represents the parameters of analogy module.

3.5 Reasoning Module

Taking the structure of the word math problem into account, we know that the operands of an expression are likely to come from the problem description X_q. Therefore, we design a reasoning module with a copy mechanism (See et al., 2017), which is build based on the last layer of analogy module. As shown in Figure 3, given a decoder state $\hat{\mathbf{x}}_e^t$, the vocabulary distribution $p_g(y_t|X_q)$ and the copy distribution $p_c(y_t|X_q)$ are formulated as follows:

$$p_g(y_t|X_q) = \frac{e^{\phi_g(y_t)}}{\sum_{y \in V} e^{\phi_g(y)}}, \tag{3}$$

$$p_c(y_t|X_q) = \frac{1}{h} \sum_{j \leq h} \sum_{i:x_i=y_t} \frac{\phi_x^j(x_i)}{\sum_{x_k \in X_q} \phi_x^j(x_k)}, \tag{4}$$

where the generated probability $p_g(y_t|X_q)$ is implemented as a fully-connected layer ϕ_g followed by the analogy module with weights W_g. And $\phi_x^j(x_i)$ indicates the j^{th} head attention value (Vaswani et al., 2017) of token x_i, h is the total number of the attention head.

To combine the vocabulary distribution $p_g(y_t|X_q)$ and the copy distribution $p_c(y_t|X_q)$, we use a learnable value p_{gen} to calculate the aggregated distribution $p(y_t|X_q)$ as follows:

$$p_{gen} p_g(y_t|X_q) + (1 - p_{gen}) p_c(y_t|X_q), \tag{5}$$

where probability p_{gen} is computed by a fully-connected layer followed by the analogy module with weights W_p. Therefore, the reasoning module can decide whether to copy the number in the problem description according to the context.

3.6 Learning Details

Suppose the length of expression of a problem is N, the goal of our model is to generate a token probability distribution $p_\theta(y_t|X_q, Z, y_{1:t-1})$ based on the problem and its retrieved question, where $t \leq N$ and θ is the parameters of the model. Next, we marginalize the token distribution to generate the t^{th} output distribution $p_\theta(y_t|X_q, y_{1:t-1})$ based on the top-K retrieved questions Z. Finally, generating each token y_t sequentially is able to form a complete expression X_e of problem X_q. Formally, the framework $p_\theta(y|X_q)$ can be defined as follows:

$$\prod_t^N \mathop{\mathbb{E}}_{Z \in top-K(p(Z|X))} p_\theta(y_t|X_q, Z, y_{1:t-1}), \tag{6}$$

where $top-K(p(Z|X))$ is a probability model that instantiated as a memory module to retrieve K similar questions. The loss function can be defined as the negative marginal log-likelihood as follows:

$$\mathcal{L}_{analogical} = -log(p_\theta(y|X_q)), \tag{7}$$

where $p_\theta(y|X_q)$ is a probability model of REAL illustrated in Eqn. (6). In order to facilitate the inductive learning of model, we further employ an auxiliary loss illustrated in Eqn. (2). Therefore, the total loss function is defined as a weighted sum of analogical loss and inductive loss. Formally, our training goal is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{analogical} + \lambda \mathcal{L}_{inductive}, \tag{8}$$

which λ is a hyperparameter for balancing the weights between $\mathcal{L}_{analogical}$ and $\mathcal{L}_{inductive}$. We simply set λ equal to 1 and found it works well in all experiments.

4 Experiments

In this section, we conduct extensive experiments on two well-known datasets to answer the following five research questions:

RQ1 How does our proposed REAL framework perform as compared to other state-of-the-art competitors?

RQ2 Are memory and copy mechanisms equally important? How does REAL model perform if one mechanism is removed?

RQ3 How does REAL perform with respect to various number of retrieved questions?

RQ4 How does REAL perform when solving problems of varying expression lengths (difficulties)?

RQ5 Can we visualize the solving process for MWP task?

4.1 Experimental Settings

4.1.1 Datasets

We evaluate our framework on two datasets, Math23K[2] (Wang et al., 2017) and Ape210K[3] (Zhao et al., 2020). The Math23K dataset labeled

[2] https://github.com/SumbeeLei/Math_EN/tree/master/data

[3] https://github.com/Chenny0808/ape210k

with equations and answers contains 22,162 questions in the training set and 1,000 questions in the testing set. Since most of the state-of-the-art results were experimented via 5-fold cross-validation and a published testing dataset, we evaluate REAL on both settings. Ape210K is a relatively large-scale dataset containing 210,488 math word problems, which are split into training, validation and testing subsets. Both validation and testing subsets have 5,000 samples and we leave the rest of 200,488 as the training samples.

4.1.2 Baselines

To justify the effectiveness of our method, we compare it to state-of-the-art baselines:

- **DNS (Wang et al., 2017).** This is a vanilla Seq2Seq model that jointly utilizes a number mapping technique and an equation template technique to generate the expression of problems.
- **Math-EN (Wang et al., 2018).** A preprocessed technique that called equation normalization is proposed to significantly reduce the template space.
- **T-RNN (Wang et al., 2019b).** This method applies a tree-structure Seq2Seq model to predict suffix expression, with inferred numbers as leaf nodes and unknown operators as inner nodes.
- **StackDecoder (Chiang and Chen, 2019).** This method proposes a stack-based decoding process to model semantic meanings of operands and operations of MWPs.
- **GTS (Xie and Sun, 2019).** This is a goal-driven tree-structured model to decode the expression in both top-down and bottom-up manners.
- **TSN-MD (Zhang et al., 2020a).** This method proposes a teacher-student networks with multiple decoders to improve the diversity of generated expressions.
- **Graph2Tree (Zhang et al., 2020b).** This method designs a graph network to enrich quantity representations and decodes the expression using a tree-based decoder like GTS.
- **Ape (Zhao et al., 2020).** This paper proposes a feature-enriched and copy-augmented Seq2Seq model, which achieves competitive performance on both Math23K and Ape210K datasets.

Model	Math23K	Math23K*	Ape210K
DNS	-	58.1	-
Math-EN	66.7	-	-
T-RNN	66.9	-	-
StackDecoder	-	65.8	52.28
GTS	75.6	74.3	56.56
TSN-MD	77.4	75.1	-
Graph2Tree	77.4	75.5	-
Ape	-	77.5	70.20
REAL	**82.3**	**80.8**	**77.18**

Table 1: The overall comparison of REAL and various methods on Math23K and Ape210K datasets. Note that Math23K denotes results on public testing set and Math23K* denotes 5-fold cross-validation. Note that the previous results evaluated on Ape210K dataset are published by Zhao et al. (2020). (Section 4.2)

4.1.3 Implementation Details

Our model is implemented based on the PyTorch[4] framework on a server equipped with 2 NVIDIA 1080Ti GPU. In the REAL model, the representation module and analogy module are both constructed by 6 layers Transformer block (Vaswani et al., 2017). To initialize the hidden layers in Transformer, we set their parameters with a pre-trained BERT (Devlin et al., 2019). The equation normalization technique (Wang et al., 2018) is applied in the training stage, which follows the previous works for fair comparison. Our model is trained for 80 epochs where the mini-batch size is set to 12. In each mini-batch, problems with their corresponding retrieved questions are randomly sampled from the training set. For optimizer, we use ADAM optimization algorithm (Kingma and Ba, 2015) with the learning rate of 5e-4, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. In addition, the learning rate is halved per 5 epochs when the total epoch is greater than 40 and we also set beam size to 5 in beam search during decoding. Lastly, we treat the predicted expression as correct if its calculated value equals to the answer, and we use the answer accuracy as the evaluation metric which follows previous works (Wang et al., 2018; Zhao et al., 2020).

4.2 Overall Performance Comparison (RQ1)

To demonstrate the effectiveness of our proposed REAL solution, we compare it to several state-of-

[4] http://www.pytorch.org

Model	Math23K*	Ape210K
REAL	80.8	77.18
w/o EN	- 0.6	- 0.16
w/o Copy	- 0.4	- 0.56
w/o Memory	- 0.9	- 0.62
w/o All	- 1.6	- 0.70

Table 2: Performance comparisons of various components on Math23K* and Ape210K datasets. (Section 4.3)

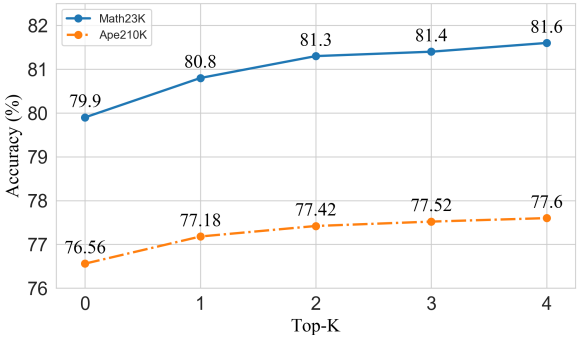

Figure 4: The performance of REAL w.r.t. various number of retrieved questions. (Section 4.4)

the-art approaches: 1) DNS; 2) Math-EN; 3) T-RNN; 4) StackDecoder; 5) GTS; 6) TSN-MD; 7) Graph2Tree; and 8) Ape.

Table 1 shows the comparison results on Math23K and Ape210K datasets among different methods, we have the following observations: 1) Our proposed REAL method shows the best performance on all benchmark datasets as compared to other methods. To verify the statistical significance of our improvement, we further conduct one-sample t-test on Math23K* experiments compared to the accuracy of Ape model and acquire a p-value about 4e-4, which unveils the superiority of our algorithm. 2) Jointly observing the experimental results on Math23K and Ape210K, we can see that our proposed model has better improvement on Ape210K dataset as compared to the improvement on Math23K. This is probably because our model is more effective on the large-scale dataset. 3) We do not perform any handcraft preprocessing steps to reduce the difficulty of model training, such as number mapping (Wang et al., 2017; Zhao et al., 2020) and relation extraction (Zhang et al., 2020b), and still achieves great performance, which manifests the effectiveness of our proposed framework.

4.3 Ablation Study (RQ2)

To evaluate the effectiveness of our proposed analogical learning method, especially the design of equation normalization technique, memory component and copy mechanism, we conduct ablation study on these components. In particular, we employ EN to denote equation normalization technique, Copy to denote the copy mechanism and w/o Memory to denote the model trained by inductive loss without using the memory module.

The performance of the three-component ablation study is shown in Table 2. We have the following observations: 1) By comparing the results of REAL and w/o EN, the performance of model is benefited from the equation normalization technique, which reveals its effectiveness in MWP task. 2) Jointly observing the performance of w/o Copy and w/o Memory models, we can infer that the Memory component is more important than the Copy component. This is mainly because the memory component is the key to perform analogical learning, which can learn the intrinsic relationships among unsolved problem and similar questions. Meanwhile, the copy component is reasonable due to it takes the structure of MWP task into account, which also results in better performance. 3) By comparing the results of w/o All with the other experiments, we find the accuracy drops significantly, proving that the three components have positive impacts on the model's performance consistently.

4.4 Impact of Retrieved Questions (RQ3)

Although REAL is trained with only a retrieved question, we still have the flexibility to adjust the number of retrieved questions at the testing stage, which can affect the model's performance. In order to show that REAL is able to solve MWPs by analogy, we test the model according to various number of retrieved questions on Math23K* and Ape210K datasets.

As shown in Figure 4, we have the following observations: 1) With the increased number of the retrieved questions, the model's performance is monotonically improved. This clearly shows that REAL model is able to master the knowledge in an analogical way, which manifests the rationality of our proposed framework. 2) It is obviously observed that when K increases from 0 to 1, the model's performance achieves significant improvement. This is mainly because the training method of the model is changed from an inductive way to an analogical way, showing the effectiveness of the memory components. 3) The performance on both datasets are relatively stable and reach their maxi-

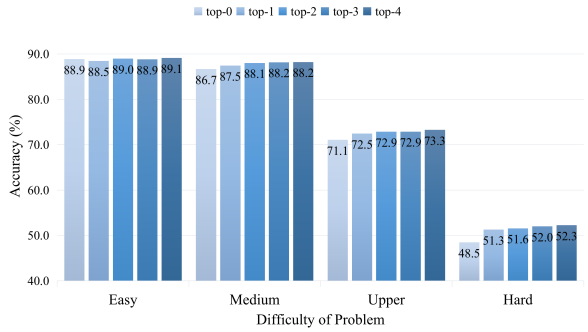

Figure 5: The top-K performance of REAL w.r.t. varying difficulties of unsolved problems. (Section 4.5)

mum values when $K = 4$. It indicates that with the increase of K, the marginal benefits of improving the model's performance will gradually diminish. We consider the noise introduced by the retrieved questions may affect the performance. Because the more questions retrieved by the memory module, the lower the similarity of the corresponding questions. 4) The experimental results demonstrate REAL's flexibility in balancing the performance and efficiency, which is an advantage of performing our memory-augmented framework in practical application.

4.5 Impact of Length (Difficulty) (RQ4)

To further evaluate REAL's analogy ability based on MWPs with different difficulty, we split every fold of Math23K dataset into 4 subsets according to the length of expression. Specifically, we deem that the longer the length of expression, the more difficult the corresponding problem is, and vice versa. Therefore, we sort the problems of the testing set according to the length of the expression in ascending order, and split it into 4 subsets, which are categorized as different difficulty levels of easy, medium, upper and hard. According to this, we conduct 20 experiments to consider how the retrieved number of questions will affect the performance under different difficulty problems. Note that each experimental result is obtained by averaging the results of the 5-fold subsets.

As shown in Figure 5, we have the following observations: 1) With the increase of difficulty, the performance of REAL gradually decreases, which is reasonable because the longer the length of expression, the more difficult for the model to predict. 2) In the "Medium", "Upper" and "Hard" experiments, the analogical results of $K \geq 1$ are noticeably superior to the inductive results of $K = 0$, which manifests the rationality of analogical learning method. Furthermore, as the number of retrieved memories increases, the model's performance is consistently improved. It demonstrates the effectiveness of our proposed analogical learning method. 3) In contrast, the experimental results are unstable when the difficulty of the problem is "Easy". We consider the reasons behind are: a) The solutions of simple problems with shorter expressions are easy to master by the model, so the model is far more likely to rely on the inductive method and can not benifit from more analogies. b) When solving relatively easy problems, the performance of the inductive-preferred model may be harmed due to the noise introduced by the increased retrieved questions. This indicates the quality of retrieved questions should be carefully considered and we leave it for future research.

4.6 Case Study (RQ5)

To better understand how the analogical learning method work in MWP task, we exploited some macro-level case studies. Specifically, we first trained a REAL model with Top-2 settings in the Math23K dataset, and selected two hard problems from the testing set that can not be solved in inductive mode but solve correctly by the analogical one.

As shown in Table 3, the case 1 describes a problem about surfacing a swimming pool by cement. The prediction is wrong when the model try to solve the problem in an inductive manner. It seems that the model is lack of common sense about the formula of cube area and misunderstands the concept of depth. To this end, we attempt to solve the problem using analogical method, which results in a correct solution. From the descriptions of problem and the retrieved questions, we can see that the REAL model is able to discover the common structure among the problem and the retrieved questions, and solve the problem through the expression template of the retrieved questions in an analogical manner. Case 2 describes a counting problem that the quantitative relationship is very complicated, in which an ingenious and complex reasoning process is required for solving the problem correctly. As shown in Table 3, it is as expected that our model fail to solve this complex problem in an inductive manner, because the existing deep learning models are still difficult to have human-like reasoning ability. In constrast, the analogical one can generate a correct solution by refering to the similar questions,

Case 1	Problem	To build a swimming pool with a length of 18 meters, a width of 10 meters, and a depth of 2 meters. We need to surface the walls and bottom of the swimming pool with cement, how many square meters of cement should be applied?
	Inductive Prediction	($18 \times 10 + 10 \times 2 + 2 \times 2$) $- 18 \times 10$ ✗
	Retrieved Questions	1) *Question:* A rectangular swimming pool is 60 meters long, 40 meters wide, and 2 meters deep. Now we need to put cement on the walls and bottom. What is the area of the cement? *Equation:* ($60 \times 40 + 40 \times 2 + 60 \times 2$) $\times 2 - 60 \times 40$
		2) *Question:* A rectangular water pool, 20 meters long, 10 meters wide, and 2 meters high. We need to surface the walls and bottom of the pool with cement. How many square meters of cement do we need to apply? *Equation:* $20 \times 10 + 20 \times 2 \times 2 + 10 \times 2 \times 2$
	Analogical Prediction	($18 \times 10 + 10 \times 2 + 18 \times 2$) $\times 2 - 18 \times 10$ ✓
Case 2	Problem	A class held a math competition with a total of 20 questions. It is stipulated that 5 points will be given for one correct answer, and 2 points will be deducted for one wrong answer. Xiao Ming got 86 points. How many questions did he answer correctly?
	Inductive Prediction	($20 \times 5 - 86$) $\div$ ($5 + 2$) ✗
	Retrieved Questions	1) *Question:* There are 20 questions in total. 7 points will be given for one correct answer, and 4 points will be deducted for one wrong answer. Wang Lei scored 74 points. How many questions did he answer correctly? *Equation:* $20 -$ ($20 \times 7 - 74$) $\div$ ($7 + 4$)
		2) *Question:* In the knowledge competition, there are 10 judgment questions. The scoring rules are: 2 points for each correct answer, and 1 point will be deducted for wrong answer. Xiao Ming only got 14 points. How many questions did he answer correctly? *Equation:* ($14 + 10 \times 1$) $\div$ ($2 + 1$)
	Analogical Prediction	$20 -$ ($20 \times 5 - 86$) $\div$ ($5 + 2$) ✓

Table 3: Two cases of REAL solving MWPs using inductive mode and analogical mode. (Section 4.6)

which demonstrates that our proposed framework is able to learn by analogy.

The above two cases qualitatively show that the memory-augmented component is an effective structure in REAL framework, which introduces an novel analogical approach for MWP task and opens a new possibility for future work.

5 Conclusion And Future Work

In this work, we propose a memory-augmented solver called REAL for MWPs. Under the REAL framework, there are four key components: 1) Memory module; 2) Representation module; 3) Analogy module; 4) Reasoning module, which are proposed to perform analogical learning schema based on the retrieved similar questions. In addition, to enhance the generation performance, a copy mechanism is designed to properly aggregate the information of operands from the problem description. The experimental results show that REAL achieves state-of-the-art performance for MWP task. Extensive micro-scope studies demonstrate the ability of REAL in learning by analogy.

In the future, we plan to extend our work in the following two directions. First, the model's performance can be further improved if the memory module of REAL model is jointly trained with the whole framework. Second, we will consider designing a more meaningful analogy module that can take the structure of question and expression into account, thus providing more information for the reasoning module to generate the problem solution.

References

Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019a. Towards knowledge-based personalized product description generation in e-commerce. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3040–3050.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2019b. Bidirectional attentive memory networks for question answering over knowledge bases. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2913–2923, Minneapolis, Minnesota. Association for Computational Linguistics.

Ting-Rui Chiang and Yun-Nung Chen. 2019. Semantically-aligned equation generation for solving and reasoning math word problems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2656–2668.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

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

Fred X Han, Di Niu, Haolan Chen, Kunfeng Lai, Yancheng He, and Yu Xu. 2019. A deep generative approach to search extrapolation and recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1771–1779.

Tom Hope, Joel Chan, Aniket Kittur, and Dafna Shahaf. 2017. Accelerating innovation through analogy mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 235–243.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

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

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2284–2293.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in neural information processing systems*.

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 Annual Meeting of the Association for Computational Linguistics*, pages 6162–6167.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4279–4285. International Joint Conferences on Artificial Intelligence Organization.

Daniel L Schwartz, Jessica M Tsang, and Kristen P Blair. 2016. *The ABCs of how we learn: 26 scientifically proven approaches, how they work, and when to use them*. WW Norton & Company.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

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

Heyuan Wang, Ziyi Wu, and Junyu Chen. 2019a. Multi-turn response selection in retrieval-based chatbots with iterated attentive convolution matching network. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 1081–1090.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to a expression tree. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1064–1069.

Lei Wang, Dongxiang Zhang, Jipeng Zhang, Xing Xu, Lianli Gao, Bing Tian Dai, and Heng Tao Shen. 2019b. Template-based math word problem solvers with recursive neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 01, pages 7144–7151.

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

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

Jipeng Zhang, Ka Wei Lee, Ee Peng Lim, Wei Qin, and Qianru Sun. 2020a. Teacher-student networks with multiple decoders for solving math word problem. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2020b. Graph-to-tree learning for solving math word problems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3928–3937.

Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. 2017. Flexible and creative chinese poetry generation using neural memory. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1364–1373.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the International Conference on Computational Linguistics*, pages 3740–3752.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1006–1014.

Xianda Zhou and William Yang Wang. 2018. Mojitalk: Generating emotional responses at scale. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1128–1137.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 372–381.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127.

Qingfu Zhu, Lei Cui, Weinan Zhang, Furu Wei, and Ting Liu. 2019. Retrieval-enhanced adversarial training for neural response generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3763–3773.