Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7468286, 9 pages
https://doi.org/10.1155/2022/7468286

Research Article

An Improved Math Word Problem (MWP) Model Using Unified Pretrained Language Model (UniLM) for Pretraining

Dongqiu Zhang[1] **and Wenkui Li**[2]

[1]*Education Science Department of Nanjing Normal University, Nanjing 210000, China*
[2]*Information Engineering Department of Suihua University, Suihua 152000, China*

Correspondence should be addressed to Dongqiu Zhang; 307642064@qq.com

Received 25 February 2022; Revised 4 June 2022; Accepted 6 June 2022; Published 14 July 2022

Academic Editor: Shahid Mumtaz

Natural Language Understanding (NLU) and Natural Language Generation (NLG) are the general methods that support machine understanding of text content. They play a very important role in the text information processing system including recommendation and question and answer systems. There are many researches in the field of NLU such as Bag of words, N-Gram, and neural network language model. These models have achieved a good performance in NLU and NLG tasks. However, since they require lots of training data, it is difficult to obtain rich data in practical applications. Thus, pretraining becomes important. This paper proposes a semisupervised way to deal with math word problem (MWP) tasks using unsupervised pretraining and supervised tuning methods, which are based on the Unified pretrained Language Model (UniLM). The proposed model requires fewer training data than traditional models since it uses model parameters of tasks that have been learned before to initialize the model parameters of new tasks. In this way, old knowledge helps new models successfully perform new tasks from old experiences instead of from scratch. Moreover, in order to help the decoder make accurate predictions, we combine the advantages of AR and AE language models to support one-way, sequence-to-sequence, and two-way predictions. Experiments, carried out on MWP tasks with 20,000+ mathematical questions, show that the improved model outperforms the traditional models with a maximum accuracy of 79.57%. The impact of different experiment parameters is also studied in the paper and we found that a wrong arithmetic order leads to incorrect solution expression generation.

1. Introduction

The basic research of natural language processing (NLP) is human-computer language interaction, which reflects human language with algorithms that can be understood by machines. NLP can perform a vast array of tasks such as text summarization, generating completely new pieces of text, and predicting what word comes next, among others. The core is a language model (LM) based on statistics. Honestly, these LMs are a crucial first step for most of the advanced NLP tasks. This paper will begin from basic LMs that can be created with a few lines of Python code and move to state-of-the-art language models that are trained using humongous data and are being currently used by the likes of Google, Amazon, and Facebook, among others. LMs are the probability distribution of a sequence of words, which can quantitatively evaluate the possibility of a string of characters. LMs are used in speech recognition, machine translation, part-of-speech tagging, parsing, optical character recognition, handwriting recognition, information retrieval, and many other daily tasks. Its ability to model the rules of a language as a probability gives great power for NLP-related tasks. The general process includes a process of predicting the back words. And then, the probabilities of all words are used to evaluate the possibility of the existence of the text. There are two types of LM: Statistical Language Models and Neural Language Models [1–4]. Statistical LMs use traditional statistical techniques like *N*-grams, Hidden Markov Models (HMM), and certain linguistic rules to learn the probability distribution of words. For example,

Mezzoudj and Benyettou [5] augment naive Bayes models with statistical n-gram language models to address the shortcomings of the standard naive Bayes text classifier. In the work of [6], they propose a fast and simple algorithm for training NPLMs based on noise-contractive estimation, a newly introduced procedure for estimating un-normalized continuous distributions. Experiment results show that the model reduces the training times by more than an order of magnitude without affecting the quality of the resulting models. The algorithm is also more efficient and much more stable than importance sampling because it requires far fewer noise samples to perform well.

However, the estimation will be difficult in practice if the text is very long. Thus, there is a simplified method: the N-grams model. In the N-grams model, the conditional probability of the word is estimated by calculating the first N words of the current word. Unigram, bigram, and trigram are the commonly used N-grams models. Typed character N-grams reflect information about their content and context. According to previous research, typed character N-grams improve the accuracy of authorship attribution [7, 8]. However, the problem of data sparseness and inaccuracy gets worse with the larger text in these models. In order to solve the problem of data sparseness when estimating probability with the N-grams model, researchers try to use neural networks to study the language model, such as UniLM and TransFormer.

This paper proposes a semisupervised approach based on UniLM, which uses unsupervised preview and supervised tuning for language processing tasks. The goal of this approach is to learn a universal representation that requires very little adaptive adjustments when migrating to various downstream tasks. The training process of the algorithm is divided into two stages: the first stage uses language modeling targets on unlabeled data to learn the initial parameters of the neural network; the second stage uses the corresponding supervised targets to adapt these parameters to the target task. Moreover, to evaluate the performance of our model in comparison with other models, we carried out a highly challenging deep QA task on a large-scale and template-rich dataset of Math Word Problems Math23K [9]. The results show it has a maximum accuracy of 79.57%.

There are three advantages and contributions of the proposed model: (1) Although there are three language model tasks in the pretraining process, we do not need to train the three models separately because the parameters of the transformer are shared. Thanks to the self-attention masking of UniLM. (2) Parameter sharing makes the learned text representation more universal because these parameters are jointly optimized with different language models. It also alleviates the problem of over-fitting on a specific language model task. (3) The proposed model is suitable for both NLU and NLG problems.

2. Related Work

In 2000, researchers first put forward the idea of neural networks to study language models [10–12]. Until 2011, Collobert and Weston [13] used a simple deep learning model to achieve SOTA results in NLP tasks such as named entity recognition NER, semantic role tagging SRL, and part-of-speech tagging POS-tagging. More and more researchers focus on the methods based on deep learning. In 2013, the word vector represented by Word2vec [14] and Pennington et al. [15] became popular. More research has explored to improve the ability of language models from the perspective of word vectors, and focused on the semantics of words and context. In 2014, Kim proposed a TextCNN [16] model based on pretrained Word2vec for sentence classification tasks. In 2016, Joulin et al. [17] proposed a simple and lightweight deep learning model for text classification: FastText. The architecture is similar to the Word2vec CBOW model proposed by Rong et al. [18]. Experiment results show that FastText can achieve a good performance with efficiency.

In addition, researchers have tried to use various mechanisms to optimize the ability of language models such as CNN, RNN, and Transormer [19, 20]. The CNN-LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. As shown in Figure 1, a common CNN-LSTM model is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. CNN-LSTM networks are well-suited to classifying, processing, and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. In a CNN, a convolution operation is used to obtain multiple feature maps. Then, it extracts key information for classification by filtering noise information through the pooling operation. Among them, pretraining combined with downstream task fine-tuning methods is the most eye-catching trend. In [21], for example, they investigate the benefits of integrating CNNs and LSTMs and report obtaining improved accuracy for Arabic sentiment analysis on different datasets. Additionally, we seek to consider the morphological diversity of particular Arabic words using different sentiment classification levels.

In AI, pretraining imitates the way human beings process new knowledge using model parameters of tasks that have been learned before to initialize the model parameters of new tasks. In this way, old knowledge helps new models successfully perform new tasks from old experience instead of from scratch. In recent years, EMLo, GPT, and BERT frequently refreshed the SOTA result [22]. For example, [23] trained a BERT language understanding model for the Italian language (AlBERTo). In particular, AlBERTo is focused on the language used in social networks, specifically on Twitter. To demonstrate its robustness, we evaluated AlBERTo on the EVALITA 2016 task SENTIPOLC (SENTIment POLarity Classification) obtaining state-of-the-art results in subjectivity, polarity, and irony detection on Italian tweets.

Transformer [24], which is based on the attention mechanism, completely abandoned CNN and RNN, and only captured the global relationship between the input and the output. As shown in Figure 2, the transformer

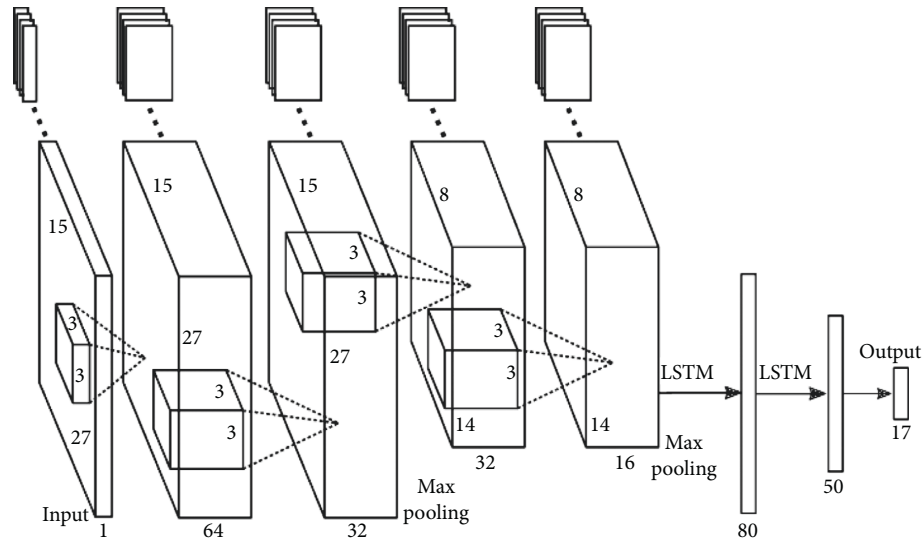

Figure 1: Illustration of the CNN-RNN-based defense architecture.

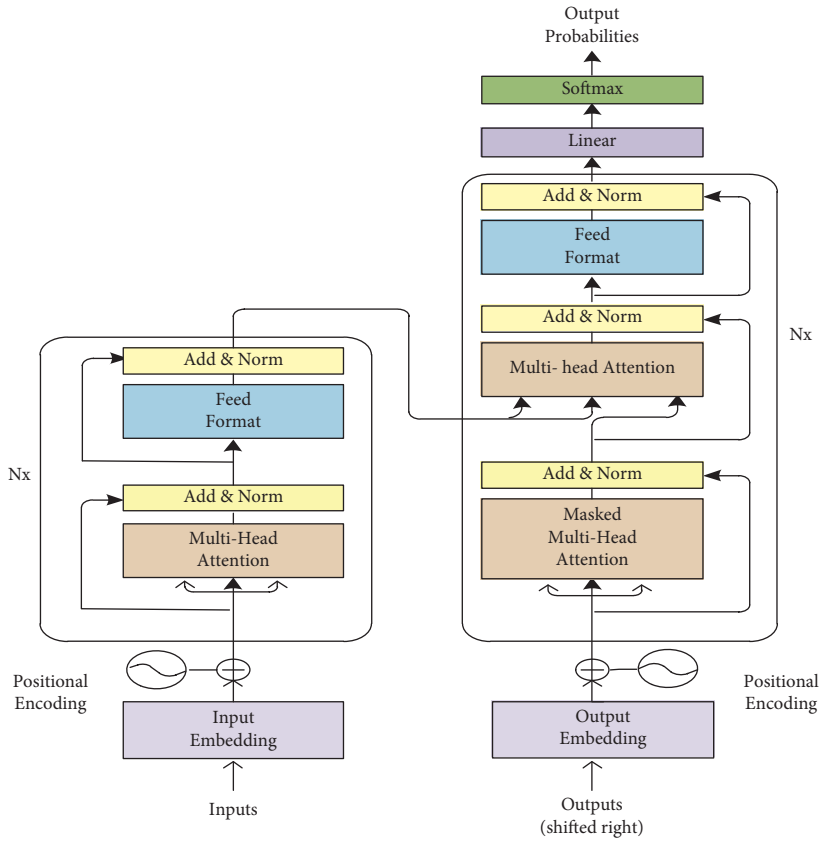

Figure 2: The transformer model architecture.

architecture is composed of two parts: Encoder and Decoder. The encoder is on the left and the decoder is on the right. Both the encoder and decoder are composed of modules that can be stacked on top of each other multiple times, which is described by N_x in the figure. We see that the modules consist mainly of multi-head attention and feed forward layers. The inputs and outputs (target sentences) are first embedded into an n-dimensional space since we cannot use strings directly.

Transformer architectures have facilitated building higher capacity models and pretraining has made it possible to effectively utilize this capacity for a wide variety of tasks. The effectiveness of transfer learning has given rise to a diversity of approaches, methodologies, and practice [25].

The framework is easier to calculate in parallel. The training time for tasks such as machine translation and parsing is reduced. Transformer's ability is obvious to all, and has been applied to pretraining models such as GPT, BERT, and XLM. In 2018, Brown et al. [26] proposed a unidirectional neural network language model GPT based on generative pretraining in OpenAI, which became one of the most popular pretraining models of the year. They use the fine-tuning method with two stages: the first stage uses the Transformer decoder, which is based on unlabeled corpus, for generative pretraining; the second stage is based on specific tasks for differentiated fine-tuning training, such as text classification, sentence pair relationship discrimination, text similarity, and multiple-choice tasks. Instead of adopting the traditional fully connected layers for classification in CNN, GPT directly feeds the resulting vector into the softmax layer.

Moreover, in 2018, Devlin et al. [25] proposed a pretraining model BERT based on a deep, two-way Transformer. Unlike GPT, the feature extractor used by BERT is the Transformer encoder part. Similarly, BERT is also divided into two stages, pretraining and downstream task fine-tuning. BERT changes the unidirectional language model in the GPT into a bidirectional one. Instead of using the standard left-to-right prediction of the next word as the target task, BERT proposes two new tasks. The first pretraining task is called MLM, or Masked Language Model. In the input word sequence of this model, 15% of the words are randomly masked and the task is to predict what they are. What we see is that, unlike previous models, BERT can predict these words from both directions—not just left-to-right or right-to-left. For example, Yu et al. [27] proposed a replication study of BERT pretraining that carefully measures the impact of many key hyper-parameters and training data size. Experimental results show that BERT achieved the SOTA results on GLUE, RACE, and SQuAD. Moreover, ERNIE [27] is an exploratory framework for continuous learning and understanding based on knowledge enhancement proposed by Baidu. The framework combines big data presets with multi-source knowledge. Through learning technology, it continuously absorbs knowledge of the text structure and learns in massive data texts to realize the model. ERNIE has achieved SOTA effects in more than 40 classic NLP missions, and has won more than 10 championships on international celebrities such as GLUE, VCR, XTREME, and SemEval.

UniLM is a BERT-based model, which is a simple but effective multimodal pretraining method of text. Unlike BERT, UniLM can be configured using different self-attention masks to aggregate context for different types of language models. It is made up of Transformer AI models jointly pretrained on large amounts of text and optimized for language modeling. The UniLM model uses three types of language modeling (one-way model, two-way model, and sequence-to-sequence prediction model) for pretraining [28]. Using a shared Transform network, a specific self-attention mask is used to control the context of prediction conditions, thereby achieving unified modeling. For example, in the work of [29], they proposes UniVL: a Unified Video and Language pretraining model for both multimodal understanding and generation. It comprises four components, including two single-modal encoders, a cross encoder, and a decoder with the Transformer backbone. Five objectives, including video-text joint, conditioned masked language model (CMLM), conditioned masked frame model (CMFM), video-text alignment, and language reconstruction, are designed to train each of the components. The train skills in [30–33] are applied in this paper.

In this paper, a semisupervised approach based on UniLM is proposed. The model allows unsupervised previewing and supervised tuning for language processing tasks. Experiment results show a maximum accuracy of 79.57% of the proposed model. The contributions of this paper as follows: this paper proposes a semisupervised way to deal with math word problem (MWP) tasks using unsupervised pretraining and supervised tuning methods, which are based on the Unified pretrained Language Model (UniLM). It combines the advantages of AR and AE language models to support one-way, sequence-to-sequence, and two-way prediction tasks. Experiments, carried out on MWP tasks with 20,000+ mathematical questions, show that the improved model outperforms the traditional models with a maximum accuracy of 79.57%.

The paper is structured as follows: we first introduce our methodology in Section 2, and then describe the test-bed and evaluate the proposed model according to several evaluation metrics in Section 3. After evaluating the performance of the proposed model, the summary and discussion about future work are described in Section 4.

3. Methodology

Researchers found that BERT could be useful for more than just Google searches [34, 35]. BERT seems to promise improvements in key areas of computational linguistics, including chat-bots, question-answering, summarization, and sentiment detection. It's defined as a "groundbreaking" technique for NLP because it's the first-ever bidirectional and completely unsupervised technique for language representation, which means a understanding of each word all at once. This represents a clear advantage in the field of context learning. It will continue revolutionizing the field of NLP because it provides an opportunity for high performance on small datasets for a large range of tasks.

The proposed model is also a multi-layer Transformer network based on UniLM, which is a BERT-based generative model. Compared to BERT, however, the proposed model can complete the three pretraining goals at the same time. Besides the mentioned pretraining methods, a new sequence-to-sequence training method is added into the model, which leads to the good performance of our model on NLU and NLG tasks. Moreover, the proposed model completes the prediction of the mask word through the context of the mask word, which is also a cloze task. For different training objectives, the context is different. The general processes of our proposed model are shown below:

(i) *Input presentation*: Each input x is a sequence composed of word tokens. The sequence can be

either a sentence or a pair of sentences combined together. The input representation is the same as UniLM. For each input token t_i, the x_i is obtained by calculating its corresponding representation through the corresponding token embedding, position embedding, and segment embedding. For the token at the beginning/end of the sequence, we add a special classification embedding (CLS)/a special end-of-sequence (SEP) of each paragraph.

(ii) *Transformer Encoder*: Then the multi-layer bidirectional Transformer encoder is used to encode the context information represented by the input. Given the input vector $X = \{x_i\}_{i=1}^{n}$, the encoding form of an L-layer Transformer's input is as follows: $H^l = \text{Transformer}(H^{l-1})$ where, $l \in [1, L], H^0 = X, H^l = [h_1^L, \ldots, h_N^L]$, and H^l is 210 the implicit vector, which is used as the contextual representation for t_i.

Pretraining Objectives: After the encoder process, we have carried out two extensions to the original UniLM pretraining goal to make full use of the rich intrasentence structure and inter-sentence structure in the language: word structure goal (mainly used for single sentence tasks) and sentence structure goal (mainly used for sentence pair tasks)). The two auxiliary targets and the original masking LM target are pretrained to find the internal language structure in a unified model. The structure is shown in Figures 3 and 4

Word Structural Objective: Figure 3 shows the method of jointly training the new word target and the mask language model target. For each input sequence, first, like UniLM, we randomly mask 15% of the token, and then send the output vector to the softmax classifier to predict the original mask. Next, given a randomly scrambled token, the order of the new words is considered. The word goal is equivalent to maximizing the possibility of placing each scrambled token in the correct position. The equation can be formulated as formula fd1:

$$\arg\max_{\theta} \sum \log P\left(\text{pos}_1 = t_1, \text{pos}_2 = t_2, \ldots, \text{pos}_k = t_k | t_1, t_2, \ldots, t_K, \theta\right). \tag{1}$$

Here, θ represents the trainable parameters in our model. K indicates the length of each scrambled subsequence. A bigger K will force the model to be able to reconstruct a longer sequence, while injecting more interference inputs. We take $K = 3$ to balance the model's reproducibility and robustness.

(iii) *Sentence Structural Objective*: The original UniLM model is very effective in predicting the next sentence (97%–98% accuracy rate). In our model, it is necessary to predict not only the next sentence but also the previous sentence, such that the pretrained language model perceives the order of sentences in a bidirectional manner. As shown in Figure 4, given a pair of sentences (S_1, S_2), where S_2 may be the next sentence of S_1 or not, probably speaking, there is a two-third probability that S_2 is the next sentence or previous sentence of S_1. Or there is a one-third probability that they are irrelevant. We use the SEP token to connect S_1 and S_2, and then the CLS encoded vector is input into the softmax classifier for the three-class prediction.

4. Experiments

In this section, we evaluate the effectiveness of the proposed model on math problems from the widely used benchmark MAWPS. MAWPS [36, 37] is an online repository of Math Word Problems and provides a unified test-bed to evaluate different algorithms. MAWPS allows for the automatic construction of datasets with particular characteristics, providing tools for tuning the lexical and template overlap of a dataset as well as for filtering ungrammatical problems from web-sourced corpora. The online nature of this repository facilitates easy community contribution. At present, the repository has amassed 3320 problems, including the full datasets used in several prominent works. Moreover, we study the effect of different parameters in our model. In the experiments, almost every possible hyper-parameter is the same for the training recipes of both models. Specifically, we carefully control the following hyper-parameters:

The same batch size: 256.

(i) The same number of training steps: $1M$

(ii) The same optimizer: Adam, learning rate $1e-4$, warmup $10K$, linear decay

(iii) The same training corpora: The dataset provides a training set containing 1674 question and answer pairs, and (251) the test set includes 865 question and answers pairs. We choose 900 questions from the 252 total training set as the development set and the remaining 1639 question and answer pairs (253) as the actual training set

(iv) The same model architecture parameters: 24 layers, 1024 hidden size, 16 heads

(v) The same fine-tuning hyper-parameter search space

4.1. Metrics. To compare the performance of different models, Macro Precision(MP), Macro Re-call(MR), and Average F1(F1) value are adopted. The final ranking is based

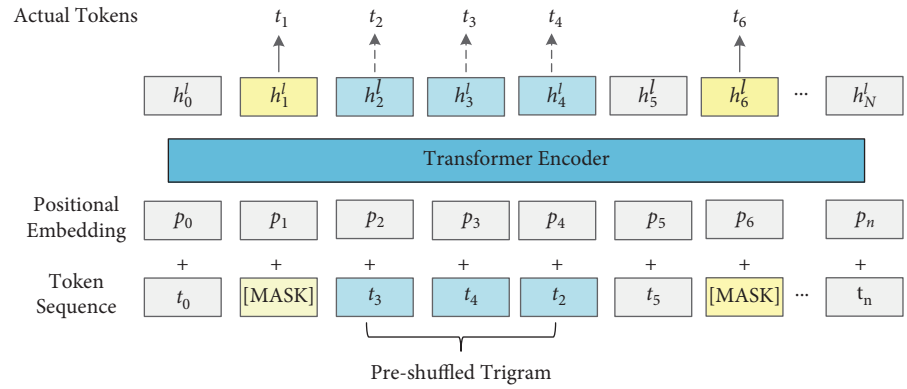

Figure 3: The architecture of the word structural objective.

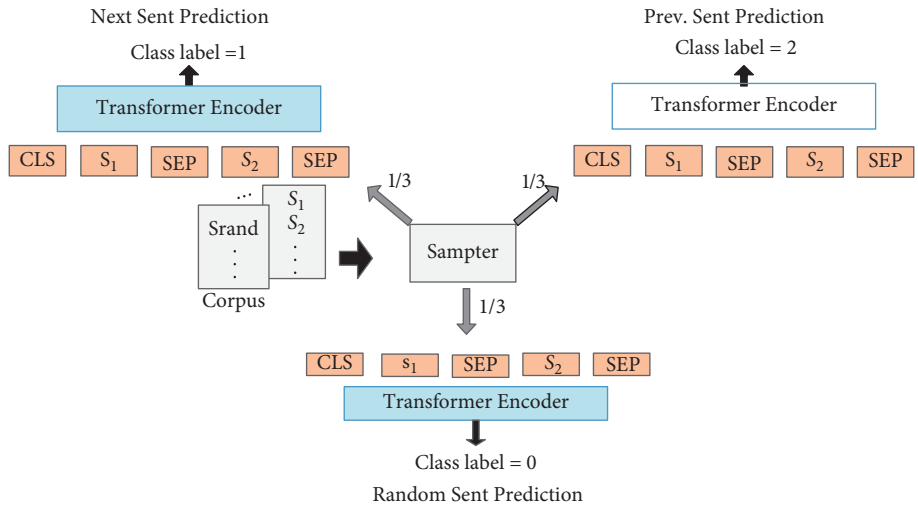

Figure 4: The architecture of the sentence structural objective.

on average accuracy. In a corpus of Q_1, Q_2, Q_3 ..., Q_N, the calculation of the three metrics is listed below:

Macro Precision(MP): As shown in formula (2), MP is the quotient of answers that are correctly selected and the total amount of dataset. MP will measure the accuracy of the model.

$$MR = \frac{|C|}{|T|}. \tag{2}$$

Macro Recall(MR): MR is the ratio of the number of shared words to the total number of words in the ground truth. As shown in formula (3), S is the amount of data that is predicted. It measures the completeness of the result.

$$MR = \frac{|C|}{|S|}. \tag{3}$$

F1: F1 score is a common metric for classification problems and is widely used in QA. It is appropriate when we care equally about precision and recall. The calculation is as shown in formula.

$$F1 = \frac{2}{MR^{-1} + MP^{-1}} = 2\frac{MR * MP}{MR + MP}. \tag{4}$$

In Table 1, it is clear that our model achieves a considerable progress in Macro Precision, Macro Recall, and F1 score. It is very hard for a model to make a huge improvement for math word problem solvers, for MWP is a mature research area.

4.2. Data Preparation. The dataset provides a training set containing 1674 question and answer pairs, and a test set including 865 question and answers pairs. We choose 900 questions from the total training set as the development set, and the remaining 1639 question and answer pairs as the actual training set.

4.3. Results. The experiment in this paper consists of two parts: Experiment 1 makes a comparison with other benchmark models. As shown in Table 1, the accuracy results of the proposed model and various baselines are listed. It is obvious that the proposed model outperforms all

Table 1: Comparison for math solving task.

Method	Macro precision	Macro recall	F1 score
Sedq2Seq	77.40	76.99	0.77
GTS	72.20	74.30	0.73
Graph2Tree	77.89	75.88	0.76
Our model	79.57	77.69	0.78

Table 2: Comparison for math solving task with different lengths of sentences.

Op	Pro	AST-Dec	GTS	Our model
1	17.4	81.5	83.1	86.2
2	51.2	74.1	79.5	84.1
3	18.4	60.1	72.1	74.2
4	6.37	43.5	52.1	53.4
5	4.30	45.6	37.6	39.4
6	0.88	56.6	46.2	56.3

Table 3: Comparison for arithmetic order errors.

Method	MWPs	Initially retrieve set
Seq2Seq	121	133
GTS	119	231
Graph2Tree	109	105
Our model	103	101

baselines in the experiments, and achieves a best accuracy of 79.57%. For example, in the experiment, the proposed method raised the F1 score to 0.78 compared with 0.73 and 0.76, respectively, of Graph2Tree and GTS [38]. This is because UniLM combines the advantages of both AR and AE models, which makes up for the disadvantages of LSTM, i.e., LSTM only stores information of one direction. Obviously, the proposed model performs the best in all tasks. To get a better understanding of how the constrained model is able to perform so well, we further carry experiments to test the effect of different parameters in our model.

4.4. Impact of the Length of the Sentence. We first study the effect of length of the sentence. The experiments are carried on the test set to investigate how the proposed model performs with increasing length of the sentence. Comparisons are built between ours and state-of-the-art models using explicit tree decoders. As shown in Table 2, we find that: First, the proposed model performs better than the other models in most of the cases, except in the case of the number of operators equals to 5. In other cases, with less than 5 operators, the model shows a good improvement compared to other models. Second, when the complexity of the sentence grows, the performance of all models decreases. This is because longer sentences lead to more complex questions, which are more difficult to predict.

4.5. Impact of Numerical Comparison. Since the wrong arithmetic order leads to incorrect solution expression generation, our proposed model aims to solve it. Experiments are carried to prove this by investigating how the model has improved the arithmetic order problem. We first retrieve the MWPs with incorrectly predicted expressions.

In the experiment, we check that the incorrectly predicted expressions length is equal to their corresponding ground truth expressions' length. As shown in Table 3, the proposed model gets 101 incorrect predicted sentences, while GTS has 119 and Graph2Tree has 103. We then check the amount of incorrectly predicted sentences with the initially retrieved set. The results show the same conclusion; our proposed model always generates fewer arithmetic order error sentences. This suggests that the proposed model is able to significantly improve the arithmetic order in MWP tasks.

5. Conclusions

This paper proposed an improved MWP model, which improves the task performance by adding UniLM for pretraining. UniLM completes unidirectional, sequence-to-sequence, and bidirectional prediction tasks. Through experiments, we show the superiority of our model against state-of-the-art models on math problem tasks.

There are three advantages of the proposed model: (1) Although there are three language model tasks in the pretraining process, we do not need to train the three models separately because the parameters of the transformer are shared. Thanks to the self-attention masking of UniLM. (2) Parameter sharing makes the learned text representation more universal because these parameters are jointly optimized with different language models. It also alleviates the problem of over-fitting on a specific language model task. (3) The proposed model is suitable for both NLU and NLG problems.

For future work, since the proposed model has difficulties dealing with long and complex sentences, we aim to consider the relationships among quantities and other attributes to better understand the context. Moreover, in future research, since advanced optimization algorithms also have been applied in many domains of NLP tasks, we may explore a comparison between advanced optimization algorithms and our model.

Data Availability

The data used to support the findings of this study are available from corresponding author upon request.

Consent

Not applicable.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Dongqiu Zhang conceptualized the study; Wenkui Li wrote, reviewed, and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Dongqiu Zhang and Wenkui Li contributed equally to this work.

Acknowledgments

Dongqiu Zhang thanks Wanli Xie for the advice given to the programming realization of this work. This work was supported by Natural Science Foundation of Heilongjiang Province under Grant LH2019F052.

References

[1] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April 2012*, vol. 80, p. 26, 2012.

[2] V. Raychev, M. Vechev, and E. Yahav, "Code completion with statistical language models," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, vol. 49, no. 6, pp. 419–428, ACM SIGPLAN Notices, New York USA, June 2014.

[3] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," 2017, https://arxiv.org/abs/1707.05589.

[4] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, *Character-aware Neural Language Models*, Thirtieth AAAI conference on artificial intelligence, Phoenix, Arizona, 2016.

[5] F. Mezzoudj and A. Benyettou, "An empirical study of statistical language models: n-gram language models vs. neural network language models," *International Journal of Innovative Computing and Applications*, vol. 9, no. 4, pp. 189–202, 2018.

[6] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," 2012, https://arxiv.org/abs/1206.6426.

[7] K. Wang, C. Thrasher, E. Viegas, X. Li, and B. j. P. Hsu, "An overview of Microsoft Web N-gram corpus and applications," in *Proceedings of the NAACL HLT 2010 Demonstration Session*, pp. 45–48, Los Angeles, California, June 2010.

[8] S. A. Taher, K. A. Akhter, and K. A. Hasan, "N-gram based sentiment mining for bangla text using support vector machine," in *Proceedings of the 2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–5, Sylhet, Bangladesh, September 2018.

[9] Y. Wang, X. Liu, and S. Shi, "Deep neural solver for math word problems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 845–854, Copenhagen, Denmark, September 2017.

[10] J. Park, X. Liu, M. J. Gales, and P. C. Woodland, "Improved Neural Network Based Language Modelling and Adaptation," in *Proceedings of the Eleventh Annual Conference of the International Speech Communication Association*, Chiba, Japan, September 2010.

[11] M. Sharma, S. Joshi, T. Chatterjee, and R. Hamid, "A Comprehensive Empirical Review of Modern Voice Activity Detection Approaches for Movies and TV Shows," *Neurocomputing*, vol. 494, pp. 116–131, 2022.

[12] X. Chen, Y. Wang, X. Liu, M. J. Gales, and P. C. Woodland, "Efficient GPU-Based Training of Recurrent Neural Network Language Models Using Spliced Sentence bunch," in *Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association*, pp. 14–18, Singapore, January 2014.

[13] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, New York, NY, USA, July 2008.

[14] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[15] J. Pennington, R. Socher, and C. D. G. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, (EMNLP), Doha, Qatar, October 2014.

[16] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *Advances in Neural Information Processing Systems*, vol. 28, pp. 919–927, 2015.

[17] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jegou, and T. Mikolov, "Fasttext.zip: compressing text classification models," 2016, https://arxiv.org/abs/1612.03651.

[18] X. Rong, "Word2vec parameter learning explained," 2014, https://arxiv.org/abs/1411.2738.

[19] S. Karita, N. Chen, T. Hayashi et al., "A comparative study on transformer vs rnn in speech applications," in *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 449–456, Singapore, December 2019.

[20] C. Raffel, N. Shazeer, A. Roberts et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, https://arxiv.org/abs/1910.10683.

[21] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," in *Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 179–191, Berlin, Germany, 2018.

[22] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," 2019, https://arxiv.org/abs/1905.05950.

[23] M. Polignano, P. Basile, D. M. Gemmis, G. Semeraro, and V. A. Basile, "Italian BERT language understanding model for NLP challenging tasks based on tweets," in *Proceedings of the 6th Italian Conference on Computational Linguistics, CLiC-it 2019. CEUR*, vol. 2481, pp. 1–6, Pisa, Italy, November 2019.

[24] N. Kitaev, L. Kaiser, and A. R. Levskaya, "The efficient transformer," arXiv preprint, 2020, https://arxiv.org/abs/2001.04451.

[25] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[26] T. B. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners," 2020, https://arxiv.org/abs/2005.14165.

[27] F. Yu, J. Tang, W. Yin et al., "Ernie-vil: knowledge enhanced vision-language representations through scene graph," vol. 1, p. 12, 2020, https://arxiv.org/abs/2006.16934.

[28] Z. Li, J. Cai, S. He, and H. Zhao, "Seq2 seq dependency parsing," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3203–3214, New Mexico, USA, August 2018.

[29] H. Luo, L. Ji, B. Shi et al., "Univl: a unified video and language pre-training model for multimodal understanding and generation," 2020, https://arxiv.org/abs/2002.06353.

[30] M. A. Dulebenets, "An Adaptive Polyploid Memetic Algorithm for scheduling trucks at a cross-docking terminal," *Information Sciences*, vol. 565, pp. 390–421, 2021.

[31] J. Li, L. Wang, J. Zhang, Y. Wang, B. T Dai, and D. Zhang, "Modeling intra relation in math word problems with different functional multi-head attentions," in *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, pp. 6162–6167, Florence, Italy, July 2019.

[32] J. Pasha, A. L. Nwodu, Fathollahi et al., "Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings," *Advanced Engineering Informatics*, vol. 52, Article ID 101623, 2022.

[33] W. Yu, Y. Wen, F. Zheng, and N. Xiao, "Improving math word problems with pre-trained knowledge and hierarchical reasoning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3384–3394, Punta Cana, Dominican Republic, November 2021.

[34] Z. Obied, A. Solyman, A. Ullah, A. Fat'hAlalim, and A. Alsayed, "BERT Multilingual and Capsule Network for Arabic Sentiment Analysis," in *Proceedings of the 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1–6, Khartoum, Sudan, March 2021.

[35] C. R. B. N. Sur, "Enhancement in language attribute prediction using global representation of natural language transfer learning technology like Google BERT," *SN Applied Sciences*, vol. 2, pp. 1–15, 2020.

[36] S. Mandal and S. K. Naskar, "Solving Arithmetic Mathematical Word Problems: A Review and Recent Advancements, Advances in Intelligent Systems and Computing," in *Information Technology and Applied Mathematics*, pp. 95–114, Springer, Singapore, 2019.

[37] L. Wang, D. Zhang, J. Zhang et al., "Template-based math word problem solvers with recursive neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7144–7151, Hawaii, Hi, USA, January 2019.

[38] H. Kitagawa, T. Takenouchi, R. Azuma et al., "Safety, pharmacokinetics, and effects on cognitive function of multiple doses of GTS-21 in healthy, male volunteers," *Neuropsychopharmacology*, vol. 28, no. 3, pp. 542–551, 2003.