Tag Embedding and Well-defined Intermediate Representation improve Auto-Formulation of Problem Description

Sanghwan Jang

Auto-Formulation of Problem Description

Auto-formulation is the task of converting an optimization problem described in natural language into a canonical representation that the optimization solver can process. In this study, a problem description and tagged entities such as variables and constraint directions are given, and the proposed method should extract the coefficients and constants of the objective and constraints for the given linear programming problem.

Data Preprocessing

Intermediate Representation and Data Augmentation

We decompose auto-formulation into two stages: (1) translating the optimization problem into intermediate representation and (2) converting the intermediate representation into canonical representation. This allows the model to only process the first stage, which eases the training of the model. We define intermediate representation to be in the form of mathematical expression (*e.g.*, 3x + 4y ≤ 50), which the pre-trained BART already knows.

The objective and constraints are generated at once so that the model can auto-formulate the optimization problem considering the relationship between mathematical expressions. To avoid inconsistency in model training, we define the order of generation of mathematical expressions and convert the labels to intermediate representation accordingly.

We augment the data by reversing the direction of some constraints (*e.g.*, replace ‘must not’ with ‘must’).

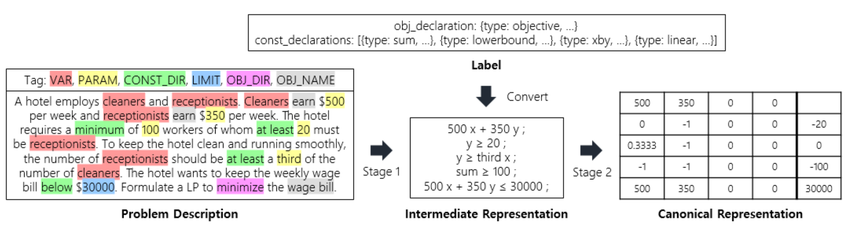

Figure 1: Two Stage Auto-Formulation.

Model

Entity Tag Embedding and Embedding Scaling

We use $\text{BART}_{\text{large}}$, which is a pretrained model for sequence-to-sequence tasks. For a given input sequence $S = [w_1, w_2, \dots, w_L]$, BART computes the token embeddings $E^{tok}_{w_1}, E^{tok}_{w_2}, \dots, E^{tok}_{w_L} \in \mathbb{R}^d$ and position embeddings $E^{pos}_1, E^{pos}_2, \dots, E^{pos}_L \in \mathbb{R}^d$, where E^{tok} is the token embedding matrix, E^{pos} is the position embedding matrix, and d is the dimensionality of BART layers. Then, the sum of token embeddings and position embeddings (*i.e.*, $E^{tok}_{w_l} + E^{pos}_l$) is forwarded to the BART encoder.

We introduce entity tag embedding to utilize a given entity tag sequence $T = [t_1, t_2, \dots, t_L]$. We compute the entity tag embeddings $E^{tag}_{t_1}, E^{tag}_{t_2}, \dots, E^{tag}_{t_L} \in \mathbb{R}^d$ and add them to the input embeddings of the BART encoder (*i.e.*, $E^{tok}_{w_l} + E^{pos}_l + E^{tag}_{t_l}$). To alleviate the destruction of pretrained knowledge, we initialize the tag embedding matrix E^{tag} to 0s.

When finetuning a pretrained model, the balance between the existing and newly introduced parameters has a significant impact on the accuracy of the model. To control this balance, we use an embedding scaling hyperparameter λ for entity tag embedding. That is, the l-th input embedding of the BART encoder is $E^{tok}_{w_l} + E^{pos}_l + \lambda E^{tag}_{t_l}$.

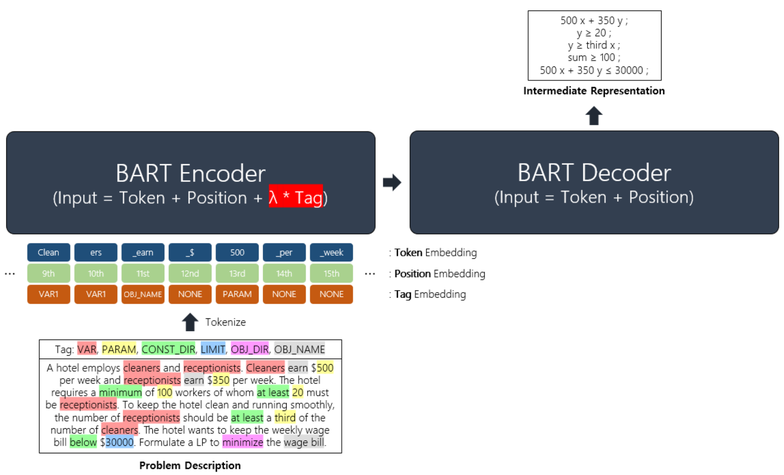

Figure 2: Model Overview.

Experiments

To evaluate the accuracy of the model, we use declaration-level mapping accuracy defined as:

$$Accuracy = 1 - \frac{\sum_{i=1}^{N} FP_i + FN_i}{\sum_{i=1}^{N} D_i},$$

where N is the number of optimization problems, D is the number of ground truth declarations (*i.e.*, objective and constraints), FP is the number of generated declarations that do not match the ground truth and FN is the number of ground truth declarations that the model failed to generate.

For the experiments, we use a batch size of 16, AdamW optimizer with a learning rate of 5e-5 and weight decay of 1e-5, and cosine annealing learning rate scheduler. We use gradient clipping with max norm of 1.0 and train the model for 100 epochs. For sequence generation, we use beam search with num beams of 4.

The results of the ablation study demonstrate that (1) using a larger pre-trained model, (2) adding entity tag embedding, (3) adjusting the weight of entity tag embedding, and (4) data augmentation improve the validation accuracy of the model. Surprisingly, just adjusting the scaling of entity tag embedding increased the validation accuracy by 8.65%, which is a huge improvement considering the simplicity of this technique.

Hyperparameter			Accuracy
BART Size	λ	p	
Base	0	0	0.5513
Large	0	0	0.7718
Large	1	0	0.8000
Large	5	0	0.8692
Large	5	0.3	0.8846

Table 1: Ablation Study.
λ is the embedding scaling weight for entity tag and p is the probability of reversing the constraint direction for data augmentation.

References

Rindranirina Ramamonjison et al. Augmenting Operations Research with Auto-Formulation of Optimization Models from Problem Descriptions. arXiv preprint arXiv:2209.15565, 2022.

Mike Lewis et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.