Appendices

Descriptive Texts for All Relations.

We use the following descriptive texts for each relation.

- PART-OF: is part of
- PART-OF-INV: has component
- USED-FOR: is used for
- USED-FOR-INV: uses
- COMPARE: compare to
- COMPARE-INV: compare to
- FEATURE-OF: is feature of
- FEATURE-OF-INV: has feature
- HYPONYM-OF: is hyponym of
- HYPONYM-OF-INV: is superordinate of
- EVALUATE-FOR: is used to evaluate for
- EVALUATE-FOR-INV: is evaluated by
- CONJUNCTION: is conjunction of
- CONJUNCTION-INV: is conjunction of

Examples of Generated Reviews

Below are two examples generated by our system using both citation graph knowledge and concept graph knowledge.

(A) Originality : The idea of smoothed rank and sorting is not new , but the paper is the first to use it in an end-to-end learning setting . Quality : The paper is technically sound and well-written . Clarity : This paper is well-organized and easy to follow . Significance : This work is significant as it provides a differentiable way to use rank/sorting operators in the context of learning to rank , which is an important problem in machine learning .

(B) This paper proposes a method to sparsify the weights and activations of a neural network by sparsifying the gradients in the backward pass . The idea is inspired by meProp , which sparsifies gradients with relatively small magnitude . The authors show that the proposed method can reduce the memory footprint by 23 % to 37 % for activations and 50 % to 80 % for weights . The paper is well written and easy to follow . The experimental results on CIFAR-10 and ImageNet are promising . However , I have the following concerns : 1 . It is not clear to me why the authors chose to use SAW instead of meProp as the baseline method . It seems to me that meProp is a better baseline than SAW . 2 .The authors claim that the memory benefit of SAW is not present for meProp since there is no storage benefit since they are temporary values generated during back-propagation . But I do n't see any evidence to support this claim . 3 .In the experiments , the authors only compare with SAW and meProp . It would be more convincing if the authors can also compare with other sparsification methods , e.g. , [1] and [2] . 4 .It would be interesting to see how the performance of the method changes with the number of epochs . 5 .It is better to provide more details on the implementation details . For example , what is the hyper-parameter used for SAW ? 6 .In Figure 3 , it would be better to show the training curves of the baselines . 7 .In Section 4.2 , it is better for the authors to provide the training curve of the baseline methods . 8 .In Table 1 , it seems that the authors did not report the training time of the proposed methods . It will be better if they can provide the running time of all the methods . 9 .In Algorithm 1 , why do the authors use the same sparsity level for all methods ? 10 .In section 4.3 , it will be helpful if the author can provide more explanations on why the sparsity is set to 50 % . 11 .It will also be helpful for the readers to provide some intuition on why this method works .

Additional Training Details

Hyperparameter Setting During *oracle pre-training*, the learning rate for GAT layers and cross attention modules are set to be $1e^{-4}$. Other components are pre-trained with a learning rate of $4e^{-5}$. We linearly increase the learning rate in the first 1/10 training steps and decrease it linearly to 0 afterward. For models using both concept graph and citation embeddings, we fine-tuned it from the model that has been pre-trained using concept graph. All models are trained for 10 epochs. For each setting, we take the one that achieves the lowest loss on the validation set to do generation.

Training Resources For models without concept graph components, we used 2 NVIDIA GeForce RTX 2080 Ti. For models with concept graph components, we used 3 NVIDIA GeForce RTX 2080 Ti. We shard the model into different devices, and the training time was about 7 hours.

Additional Evaluation Details

We observe that generated reviews are highly structured and follow a specific routine : (i) they first describe the core idea of a paper, (ii) then followed by enumerations of strengths and weaknesses. The clear organization makes it possible for us to use some heuristic rules to decide the sentiment of a specific review by simply looking into different aspects with sentiment polarity that are mentioned. When calculating recommendation accuracy, we first annotate 50 samples in the test dataset and then conclude some heuristic rules to automatically label the sentiment of a review. The rules are:
If a review contains more than or equal to three positive aspects, then we regard its sentiment to be positive.
If the number of positive aspects is more than or equal to the number of negative aspects in a review, then we regard its sentiment to be positive.
If there is only one positive aspect or there are no positive aspects identified in a review, then we regard its sentiment to be negative.
Other cases are identified as Neutral.

We also ask a human annotator to annotate the sentiment of 50 randomly selected system-generated reviews. The cohen kappa between human annotator and system annotator is 0.4291, which stands for moderate agreement.