RelTransformer: A Transformer-Based Long-Tail Visual Relationship Recognition Supplementary

Jun Chen¹, Aniket Agarwal², Sherif Abdelkarim¹, Deyao Zhu¹, Mohamed Elhoseiny¹
¹King Abdullah University of Science and Technology
²Indian Institute of Technology
{jun.chen,deyao.zhu,mohamed.elhoseiny}@kaust.edu.sa
aagarwal@ma.iitr.ac.in, sherif.abdelkarim91@gmail.com

A. Training and Implementation Details

To have fair comparisons with all the baseline models, we follow the same experimental setup as [1] for GQA-LT and VG8K-LT evaluation and also the same setup as [10] for VG200 evaluation. We train our model and all its ablations with 8 V100 GPUs. The batch size is 8. We train our models with 12, 8, and 7 epochs on GQA-LT, VG8K-LT, and VG200 datasets, respectively. The hidden size of \( h \) is 768. The number of Transformer heads is 12. Relational and global-context encoders both have 2 layers. The memory size is \( 100 \times 768 \). We use the Faster R-CNN [6] with VGG-16 [7] backbone to extract the object proposal features. We also apply the pretrained Word2Vec [5] embeddings to represent the relation and object labels.

B. Subject and Object Per-Class Accuracy

We also provide the per-class accuracy for subjects and objects in Table 1. We can observe that our RelTransformer can also easily outperform many baselines. Combining RelTransformer with CE loss can outperform all the baseline models on each category and it significantly improves over the “many” category. Combining it with WCE can furthermore improve over “medium” and “tail” categories on both datasets. These results demonstrate the effectiveness of our model on the subject and object prediction. We also notice that RelTransformer (WCE or DCPL) drops the performance over “many” categories compared to CE loss function; We hypothesis that it is due to the assigned lower weights on the high-frequent subjects/objects, and lower weights will lead to the lower confidence values during the classification. This phenomenon can explain why RelTransformer (WCE) underperforms on the “many” classes and why it underperforms on the compositional prediction for “many” and “medium” classes.

C. Per-Example Accuracy

We provide the per-example accuracy for the subjects and objects on GQA-LT dataset. It shows that RelTransformer (CE) achieves the best performance on predicting subjects/objects and relations among all the baselines. Per-example accuracy is mainly dominated by the “head” classes since they are very large in example numbers. This indicates that RelTransformer (CE) improves both head and tail classes. Compared to the best baseline LSVRU (CE), RelTransformer (CE) improves it by 10.6 acc on subject/object and 0.3 acc on relation predictions. However, we could also notice that RelTransformer (WCE or DCPL) brings the performance down on per-example accuracy, and this phenomenon is also observed in LSVRU baselines. But the results in Table 1 (Supplementary) and Table 1 (main paper) shows that RelTransformer (WCE or DCPL) has a
Table 1. Average per-class accuracy for subject/object. We separately evaluate the average per-class accuracy for many, medium, few, and all categories. The best performance from each category is underlined. ♯ denotes our reproduction. Our model is denoted in gray.

Table 2. Per-example Accuracy on GQA-LT. The best performance from each category is underlined. Our model is denoted in gray.

D. Memory Attention Further Analysis

To further investigate the role of memory attention, we compute the memory attention scores, \( J - \alpha \) in our fusion function Eq. 1, for each testing example from well-trained RelTransformer (WCE) on the GQA-LT dataset. We average the attention scores per each relation-class and demonstrate them in Fig. 1. The relation classes are ranked according to their frequency in the training data. We can observe a clear increasing trend for the attention scores from high-frequent relations to low-frequent ones, meaning that the features from memory can contribute to the long-tail relations more than the head-relations, which can reflect why memory can gain more improvement on the “medium” and “tail” classes.

\[
g(x, y) = \alpha \odot x + (J - \alpha) \odot y
\]

where \( W \) is a 2D \( \times \) D matrix. \( b \) is a bias term. \([;]\) denotes the concatenation. \( \odot \) is the Hadamard product. \( J \) is an all-one matrix with the same dimensions as \( \alpha \).

E. Additional Qualitative Examples

We show the relation prediction results on VG200 dataset in Fig. 2 and provide more long-tail relation prediction results on GQA-LT and VG8K-LT dataset in Fig. 3 and 4.
Figure 2. Qualitative results on VG200 dataset
Figure 3. More long-tail relation prediction examples on GQA-LT dataset.
Figure 4. More long-tail relation prediction examples on VG8K-LT dataset
References


