Supplementary Materials of "Document-level Relation Extraction as Semantic Segmentation"

1 Implementation Details

1.1 Encoder Module

Given the document $d = [x_t]_{t=1}^{L}$, we insert special symbols "$< e >$" and "$< /e >$" at the start and end of mentions to mark the entity positions. We leverage the bert-base-cased and roberta-large for the encoder of DocuNet-BERT$_{\text{base}}$ and DocuNet-RoBERTa$_{\text{large}}$ respectively to obtain the embedding of document. Considering some documents are longer than 512, we thus leverage a *dynamic window* to encode whole documents, which average the embeddings of overlapping tokens of different windows with the fixed length of 512 to obtain the final representations. For each entity e_i with mentions $\{m_j^i\}_{j=1}^{N_{e_i}}$, we leverage a smooth version of max pooling, namely, logsumexp pooling [Jia *et al.*, 2019], to obtain the entity embedding. The entity-level relation matrix can be calculated by *similarity-based* method and *context-based* method which are described concretely in our paper. For *similarity-based* method, we directly apply three kinds of similarity functions to obtain the 3-dims feature vector based on two entities. For *context-based* method, the initial feature vector is 768-dims, which is too large for subsequent modules. Thus, we adopt MLP to compress feature vector information to 3 dimensions. To sum up, we generate the representation of entities with 768-dims and entity-level relation matrix with 3 channels through the Encoder Module.

1.2 U-shaped Segmentation Module

This module is formed as a U-shaped structure, which contains two down-sampling blocks and two up-sampling blocks with skip connections. On the one hand, each down-sampling block has two subsequent max pooling and separate convolution modules. Further, the number of channels is doubled in each down-sampling block. As the module's input is an entity-level relation matrix with 3 channels, the number of channels are respectively [3, 256, 512, 256, 128, 256] in the entire module sequence.

1.3 Classification Module

Given the entity pair embedding $\mathbf{e}_s$ and $\mathbf{e}_o$ with the entity-level relation matrix Y, we respectively concatenate the specified feature vector in matrix with the embedding of $\mathbf{e}_s$ and $\mathbf{e}_o$, and map them to hidden representations z with a feed-forward neural network. Then, we obtain the probability of relation via a bilinear function.

2 Experiments Details

Our code is available in the supplementary materials for reproducibility. We detail the training procedures and hyperparameters for each of the datasets. We utilize Pytorch [Paszke *et al.*, 2019] to conduct experiments with one NVIDIA V100 16GB GPU. All optimization was performed with the Adam optimizer with a linear warmup of learning rate over the first 6% of steps, then linear decay over the remainder of the training. Gradients were clipped if their norm exceeded 1.0, and weight decay on all non-bias parameters was set to 5e-4. The grid search was used for hyperparameter tuning (maximum values bolded below), using five random restarts for each hyperparameter setting for all datasets. Early stopping was performed on the development set. The batch size was 4 in all cases.

DocRED. The DocRED dataset is available in https://github.com/thunlp/DocRED. The hyper-parameter search space was:

- epoch: 30
- batch size: 5
- accumulate: 1
- bert learning rate: [1e-5, 2e-5, **3e-5**, 4e-5]
- unet learning rate: [2e-4, 3e-4, **4e-4**, 5e-4]
- warmup rate: 0.06

CDR and GDA. The CDR and GDA datasets can be obtained following the instructions in https://github.com/fenchri/edge-oriented-graph

CDR. The hyper-parameter search space was:

- epoch: 30
- batch size: 4
- accumulate: 1
- bert learning rate: [1e-5, 2e-5, **3e-5**, 4e-5]
- unet learning rate: [2e-4, 3e-4, **4e-4**, 5e-4]
- warmup rate: 0.06

GDA. The hyper-parameter search space was:

- epoch: 10
- batch size: 4
- accumulate: 4
- bert learning rate: [1e-5, 2e-5, **3e-5**, 4e-5]
- unet learning rate: [2e-4, **3e-4**, 4e-4, 5e-4]
- warmup rate: 0.06

References

[Jia *et al.*, 2019] Robin Jia, Cliff Wong, and Hoifung Poon. Document-level n-ary relation extraction with multiscale representation learning. In *NAACL-HLT*, 2019.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.