A Details of Hyper-parameters

We search the best hyper-parameters based on F1 on the development set. Generally, for all of CFER-GloVe, CFER-BERT$_{\text{Base}}$, CFER-RoBERTa$_{\text{Large}}$ and CFER for CDR, we use AdamW with $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\epsilon = 1e-6$, weight decay = 0.0001 as the optimizer, apply exponential moving average on all parameters with a decay rate of 0.9999, use ReLU as the activation function, and use the DCGCN consisting of two blocks with 4 sub-layers in each block. We adopt a slanted triangular scheduling strategy for learning rate, which first linearly increases the learning rate from 0 to the peak value in the first 10% steps (warm-up steps), and then linearly decreases it to 0 in remaining steps. For other key hyper-parameters, we state the values tried and the finally selected value for four models separately as follows.

For CFER-GloVe: (1) We search the peak learning rate for all modules in $\{1e-3, 1e-4\}$, and finally choose $1e-3$. (2) We search the batch size in $\{8, 16, 32\}$, and finally select 16. (3) We search the dropout rate for DCGCN modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.4. (4) We search the dropout rate for other modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.2. (5) We set the embedding dimension to 300, the same as the dimension of used GloVe embeddings. (6) We search the hidden size in $\{100, 300, 512\}$, and finally select 300. (7) For each hyper-parameter configuration, we train 300 epochs and select the best F1 achieved during these 300 epochs to evaluate the performance under this configuration.

For CFER-BERT$_{\text{Base}}$: (1) We search the peak learning rate for BERT modules in $\{1e-4, 5e-5, 1e-5\}$, and finally select $1e-5$. (2) We search the peak learning rate for the other modules in $\{1e-3, 5e-4, 1e-4\}$, and finally select $1e-3$. (3) We search the batch size in $\{8, 16, 32\}$, and finally select 32. (4) We search the dropout rate for DCGCN modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.6. (5) We search the dropout rate for other modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.2. (6) We search the hidden size in $\{300, 512, 768\}$, and finally select 512. (7) For each hyper-parameter configuration, we train 300 epochs and select the best F1 achieved during these 300 epochs to evaluate the performance under this configuration.

For CFER-RoBERTa$_{\text{Large}}$: (1) We search the peak learning rate for RoBERTa modules in $\{1e-4, 5e-5, 1e-5\}$, and finally select $1e-5$. (2) We search the peak learning rate for the other modules in $\{1e-3, 5e-4, 1e-4\}$, and finally select $1e-3$. (3) We search the batch size in $\{8, 16, 32\}$, and finally select 32. (4) We search the dropout rate for DCGCN modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.6. (5) We search the dropout rate for other modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.2. (6) We search the hidden size in $\{512, 768, 1024\}$, and finally select 1024. (7) For each hyper-parameter configuration, we train 300 epochs and select the best F1 achieved during these 300 epochs to evaluate the performance under this configuration.

For CFER for CDR: (1) We search the peak learning rate for BioBERT modules in $\{1e-4, 5e-5, 1e-5\}$, and finally select $1e-5$. (2) We search the peak learning rate for the other modules in $\{1e-3, 5e-4, 1e-4\}$, and finally

select $1e-4$. (3) We search the batch size in $\{4, 8, 16\}$, and finally select 4. (4) We search the dropout rate for DCGCN modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.6. (5) We search the dropout rate for other modules in $\{0.2, 0.4, 0.6\}$, and finally select 0.2. (6) We search the hidden size in $\{512, 768, 1024\}$, and finally select 1024. (7) For each hyper-parameter configuration, we train 100 epochs and select the best F1 achieved during these 100 epochs to evaluate the performance under this configuration.