A Supplementary Material

A.1 Dataset and Preprocessing

the JCT Dataset. We built the JCT dataset to train the data-to-text module of the medical report generation system. For the JCT dataset, we collected 4,454 medical reports regarding pulmonary nodules from a hospital. To train an accurate medical report generation system, we focused only on the findings in the reports and excluded the sentences that violated patient privacy. During a consultation with radiologists, we defined 57 types of finding labels. As preprocessing, all descriptions that were not related to any findings were truncated by annotators. We lexicalized phrases referring to the existence of nodules and phrases referring to the size of the nodules to improve the stability of training of the data-to-text generation model. We used MeCab [1] and mecab-ipadic-NEologd (Sato et al., 2017) to tokenize the reports, and keep tokens with 2 or more occurrences.

To prevent data leakage in validation/test datasets, we split the dataset in a way to ensure that the same sets of finding labels are not included in the training, validation, and test data. Additionally, to avoid the negative influence of the imbalanced frequency of sets of finding labels, we omitted the samples with duplicated sets of finding labels in the validation/test dataset. These strategies for data splitting and duplicate input handling caused differences in average labels and lengths, as shown in Table 5. If samples contained shorter sentences and fewer input labels, the validation and test datasets tended to contain longer sentences and a greater number of input labels.

the MIMIC-CXR Dataset. Medical reports in the MIMIC-CXR dataset [2] contain descriptions that are irrelevant to the findings in the input images. Hence, we extracted the finding sections of the reports using the scripts provided in Boag et al. (2019) [3]. In training data, we truncated the sentences in the reports that were not related to any findings using CheXpert Labeler and NegBio (Peng et al., 2018) parser to improve the stability of training the model. We omitted the reports that did not mention any findings or had no finding sections from the training data. Note that the reports in the validation and test data may contain a description that does not mention any findings. We use this approach to align our experimental conditions with previous end-to-end research Boag et al. (2019). We used the Natural Language Toolkit [4] to tokenize the reports, and keep tokens with 10 or more occurrences. We have split the dataset into train, validation, and test data based on the split distributed in the MIMIC-CXR-JPG (Johnson et al., 2019) [5] dataset. Table 5 presents the statistics of the MIMIC-CXR dataset.

	Number of Reports	Average labels	Average length
the JCT dataset			
Training data	3,637	4.71	27.5
Validation data	418	9.46	52.7
Test data	399	9.49	51.4
the MIMIC-CXR dataset			
Training data	131,016	4.92	43.6
Validation data	1,156	4.90	44.5
Test data	2,299	5.01	54.7

Table 5: Statistics of the JCT dataset and the MIMIC-CXR dataset.

dataset	JCT	MIMIC-CXR
Data-to-Text Module Hyperparameters		
Vocabulary size	339	2222
Number of labels	57	40
Dropout rate	0.2	0.2
Word embedding size	32	64
Label embedding size	16	16
Hidden size	32	32
Beam search width	5	5
Training Hyperparameters		
Batch size	32	32
Optimizer	Adam	Adam
Learning rate	5.0×10^{-3}	2×10^{-4}
Learning rate decay	0.99	0.98
λ_{rouge}	0.2	0.2
λ_{rl}	0.2	0.03
λ_{aug}	0.1	0.05
τ (Softmax temperature)	0.5	0.4
Dropout	0.2	0.2
Gradient clipping	2.0	2.0

Table 6: List of hyperparameters of the data-to-text modules.

A.2 Training Details

Image Diagnosis Module All images were fed into a network with a size of 512 × 512 pixels. We set up the loss as the sum of the multi-class cross-entropy for each observations and used the RAdam (Liu et al., 2019b) optimizer with a learning rate of 1.0×10^{-4}. We trained the model for 5 epochs with the CheXpert dataset (Irvin et al., 2019).

[1] https://taku910.github.io/mecab/

[2] https://physionet.org/content/mimic-cxr/2.0.0/

[3] https://github.com/wboag/cxr-baselines

[4] https://www.nltk.org/

[5] https://physionet.org/content/mimic-cxr-jpg/2.0.0/

Labels	Negative	Positive	Uncertain	No_Mention
No_Finding	-	0.468	-	0.907
Enlarged_Cardiomediastinum	0.436	0.197	0.040	0.858
Cardiomegaly	0.209	0.525	0.013	0.873
Lung_Opacity	0.002	0.696	0.000	0.602
Lung_Lesion	0.150	0.246	0.092	0.936
Edema	0.223	0.615	0.254	0.740
Consolidation	0.489	0.215	0.254	0.740
Pneumonia	0.008	0.163	0.278	0.883
Atelectasis	0.002	0.333	0.325	0.713
Pneumothorax	0.458	0.513	0.000	0.770
Pleural_Effusion	0.524	0.759	0.036	0.639
Pleural_Other	0.335	0.217	0.165	0.963
Fracture	0.234	0.207	0.007	0.890
Support_Devices	0.046	0.844	0.007	0.771
Overall F1-Score	0.240	0.428	0.103	0.807

Table 7: Evaluation of the image diagnosis module for each finding label. All scores are measured by F-score in 5-fold cross validation.

Dataset	JCT	MIMIC-CXR
Optimizer	AdamW	AdamW
Learning rate of BERT layer	2.0×10^{-5}	2.0×10^{-5}
Learning rate of FC layer	2.0×10^{-3}	1.0×10^{-4}
CBL β (Cui et al., 2019)	0.999	0.999
Warm up steps	200	200

Table 8: List of hyperparameters of the reconstructor modules.

Subsequently, we evaluated the image diagnosis module with the CheXpert dataset. To evaluate the accuracy of image classification correctly for the infrequent labels, we performed a 5-fold cross-validation. Table 7 presents F-scores for each finding labels evaluated in 5-fold cross-validation. Although the F-scores of the no-mention labels are high, the F-scores of the positive, negative, and uncertain finding labels are relatively low. This is because the CheXpert dataset is significantly imbalanced, and almost all finding labels in the training data are in the no-mention category.

Data-to-Text Module For the JCT and MIMIC-CXR datasets, we trained the data-to-text module for 50 and 20 epochs, respectively. We used a CRS score of the validation data as the stopping criteria. Finally, we reported evaluation scores that achieved the highest CRS score on the validation data. Table 6 presents hyperparameters used to train our models. Before we trained the model with RL, we pretrained the model with only cross-entropy loss for an epoch. The number of parameters of the data-to-text module was 127k for the JCT dataset and 463k for the MIMIC-CXR dataset.

Reconstructor Module To train the reconstructor for the JCT dataset, we used the pretrained Japanese BERT model [6]. We have split the training data of the data-to-text module into 4:1 and used the former part as training data and the latter part as validation data for the reconstructor. For fine-tuning, we used the AdamW optimizer with a learning rate of 2.0×10^{-5} for the BERT layer and 2.0×10^{-3} for the fully connected layer. We used binary cross-entropy loss to train the model, and applied Class Balanced Loss (CBL) (Cui et al., 2019) with $\beta = 0.999$. The number of parameters of the reconstruction module is 110M. We fine-tuned the model with 10 epochs, and the F-score on the validation dataset was 90.3.

To train the reconstructor for the MIMIC-CXR dataset, we use the pretrained bert-base-uncased model. We also verified the BioBERT model (Lee et al., 2020), but the results showed no significant differences with the bert-base-uncased model. For fine-tuning, we used the AdamW optimizer with a learning rate 2.0×10^{-5} for the BERT layer and 2.0×10^{-3} for the fully connected layer. By analogy with the JCT dataset, we have split the training data into 4:1 and used the former part as the training data and the latter part as the validation data for the reconstructor. We used binary cross-entropy loss to train the model, and applied Class Balanced Loss (CBL) (Cui et al., 2019) with $\beta = 0.999$. The number of parameters of the reconstruction module was 109M. We fine-tuned the model with 10 epochs, and the F-score on the validation dataset was 97.9.

We used an Intel Core i7-6850K CPU and NVIDIA GTX 1080Ti GPU for training on the

[6] https://github.com/cl-tohoku/bert-japanese

JCT dataset, and the training time was approximately 3 h. We used an Intel Xeon Gold 6148 CPU and NVIDIA Tesla V100 GPU for training on the MIMIC-CXR dataset, which required approximately 12 hours.

A.3 Evaluation Settings.

We use an approximate randomization test [7] to evaluate the statistical significance.

Evaluation Metrics on the JCT Dataset. For automatic evaluation on the JCT dataset, we used BLEU (Papineni et al., 2002), F-scores of ROUGE-L (Lin, 2004), and CRS as metrics. We used Natural Language Toolkit [8] to calculate BLEU scores, and the ROUGE Python library [9] to calculate ROUGE-L scores.

Evaluation Metrics on the MIMIC-CXR Dataset. For comparison with the previous image captioning approaches, we used BLEU-1, BLEU-2, BLEU-3, and BLEU-4 metrics calculated by the nlg-eval [10] library. However, word-overlap based metrics, such as BLEU, fail to assume the factual correctness of generated reports. We compared the labels assigned in CheXpert Labeler between the generated reports and gold reports to calculate the CheXpert accuracy, precision, micro F-score, and macro F-score. The micro F-score was obtained by the overall numbers of true positives, false positives, and false negatives. The macro F-score was obtained by the average of F-scores per class label. Although the micro F-score neglects infrequent labels, the score is significantly biased by the imbalanced distribution of the test dataset.

Note that precision and F-score are preferred to evaluate the clinical correctness of the reports in CheXpert. In contrast, CheXpert accuracy does not quantify the clinical correctness of the generated reports adequately. The imbalanced dataset results in an excessive number of true negatives rather than true positives. Hence, CheXpert accuracy overestimates the clinical correctness of generated reports if the reports comprise many descriptions that are not related to the findings.

Modification Flow We apply the modification process to the image diagnosis module result with the parameters of $(p_{th}^{low}, p_{th}^{high}) = (0.1, 0.9)$ for the positive finding labels. However, we regard all negative and uncertain labels predicted by the image diagnosis module as unreliable. This is because negative or uncertain findings are highly dependent on the radiologist's judgment.

[7] https://github.com/smartschat/art

[8] https://www.nltk.org/

[9] https://github.com/pltrdy/rouge

[10] https://github.com/Maluuba/nlg-eval