RadiBERT: Pre-trained BERT Model on Radiology Reports with Labeled Named Entity Recognition and Classification for Patients' Potential Examinations and Treatments

Zesheng Jia
Faculty of Science

Abstract

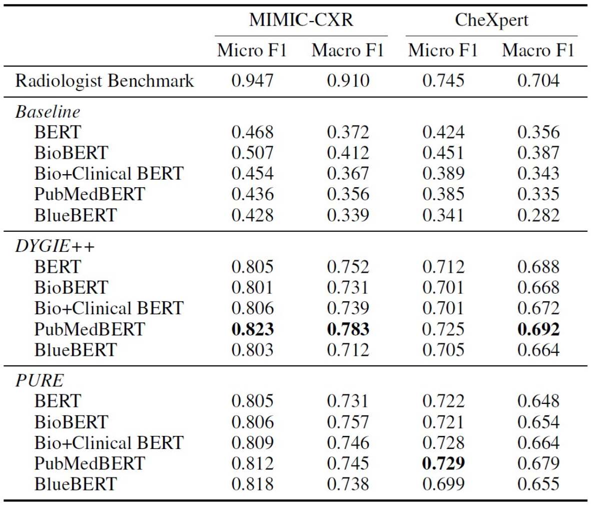
The medical resources are becoming increasingly scarce due to the pandemic and the staff shortage of hospitals workers. Doctors are facing more cases than they can handle, which is causing missed diagnosis and leading to more overtime working. The raising of deep learning model of NLP brings us the potential to solve this task even if we don't have the corresponding medical domain linguistic knowledge. In this paper, we will train and modify the original Transformers pre-trained model BERT on our radiology domain dataset for doing two main downstream tasks - Named Entity recognition task and examinations prediction. We compared the performance of BERT-base model with whether trained or not trained tokenizer on a publicly NER dataset. And we also compare the result with other people's medical BERT model on the same dataset. On the examinations prediction task, we tried two different methods. Those are multi-class and multi-label prediction BERT model. Overall, we got an 74% accuracy by the multi-label model and 59% by the multi-class model. In the meantime, the normal BERT model only has 42% accuracy.

1 INTRODUCTION

In the original BERT model and other fine-tuning BERT models, we often use a similar BERT tokenizer for different downstream tasks, such as the question-answering task on Wikipedia dataset like SQuAD [2], or the NER problem on newspaper datasets like CoNLL-2003 [3]. The results of them are relatively satisfying. But if we use those pre-trained models with the original or similar BERT tokenizer on the datasets that are in very different domains like medical reports or science articles, the result will become worse [1]. However, using a new-trained contextualized word embedding for the in-domain tasks has been proven to be a success [4].

2 RELATED WORK

In 2021, Rasmy's team created MED-BERT [2] that was published on Nature. They adapted the original BERT model with their own input Embeddings for solving the EHR health records for disease prediction. They add one Med-BERT Embedding layer on the top of medical reports input before feeding to the transformers



	MIMIC-CXR		CheXpert	
	Micro F1	Macro F1	Micro F1	Macro F1
Radiologist Benchmark	0.947	0.910	0.745	0.704
Baseline				
BERT	0.468	0.372	0.424	0.356
BioBERT	0.507	0.412	0.451	0.387
Bio+Clinical BERT	0.454	0.367	0.389	0.343
PubMedBERT	0.436	0.356	0.385	0.335
BlueBERT	0.428	0.339	0.341	0.282
DYGIE++				
BERT	0.805	0.752	0.712	0.688
BioBERT	0.801	0.731	0.701	0.668
Bio+Clinical BERT	0.806	0.739	0.701	0.672
PubMedBERT	**0.823**	**0.783**	0.725	**0.692**
BlueBERT	0.803	0.712	0.705	0.664
PURE				
BERT	0.805	0.731	0.722	0.648
BioBERT	0.806	0.757	0.721	0.654
Bio+Clinical BERT	0.809	0.746	0.728	0.664
PubMedBERT	0.812	0.745	**0.729**	0.679
BlueBERT	0.818	0.738	0.699	0.655

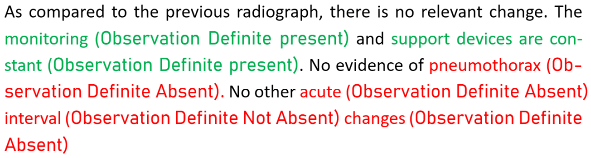
Figure 1. Radgraph NER performance

The table from Radgraph [1] that shows in-domain specific trained BERT has better performance than original and other BERT.

model. At the end, they get a overall 85% accuracy on the disease prediction. For the medical domain, W. Yu's team [11] discovered that if we pre-trained the dictionary or tokenizer before training the model, we could have higher performance on the prediction than the original BERT. Based on those two main inspirations. We will solve our own task as the following.

3 PROBLEM DEFINITION

For the NER task, if we take one paragraph from a radiology reports such as "As compared to the previous radiograph, there is no relevant change. The monitoring and support devices are constant. No evidence of pneumothorax. No other acute interval changes." [9] We want to mark the observations that are related to symptoms that are definitely happens for further examinations or some symptoms that are definitely not represented for eliminating the possibility for certain diseases.



As compared to the previous radiograph, there is no relevant change. The monitoring (Observation Definite present) and support devices are constant (Observation Definite present). No evidence of pneumothorax (Observation Definite Absent). No other acute (Observation Definite Absent) interval (Observation Definite Not Absent) changes (Observation Definite Absent)

Figure 2. NER Example

Example of after applying NER of BERT model

By extracting these two types of observations in one sentences, we can merge all those observations into forms for helping doctors to get the important information instantly.

And for the examinations prediction, we have the overall examinations of each radiology reports as the targe label. This task is solving on a confidential and private dataset that are not published yet. We would like to use the text reports to predict the corresponding patient's examinations types for predicting the diagnosis with other numerical values in the future.

4 DATASET

We will use Radgraph dataset [1] for second-time pretrained BERT model. It is a radiology reports dataset that was created in 2021 by Stanford. It contains "contains board-certified radiologist annotations for 500 radiology reports from the MIMIC-CXR dataset [10] (14,579 entities and 10,889 relations), and a test dataset, which contains two independent sets of board-certified radiologist annotations for 100 radiology reports split equally across the MIMIC-CXR and CheXpert datasets."

An entity is a continuous span of text that includes one or more continuous words. Radgraph [1] has four entities, which are labeled as "ANAT-DP" (Anatomy Definitely present), "OBS-DP" (Observation

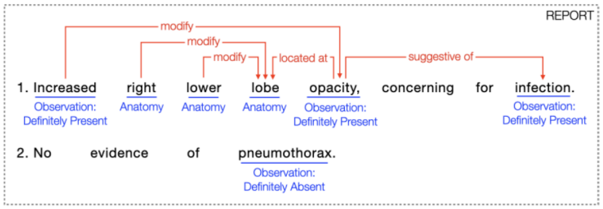


Figure 3. Radgraph annotated example

Radgraph Dataset has radiology domain annotated entities and the corresponding relations with certain groups of entities

Definitely present), “OBS-U” (Observation uncertain), and “OBS-DA” (Observation Definitely Absent). And three relations between entities, which are labeled as “suggestive of”, “located at”, and “modify”. In Figure 3, “Increased” (Observation Definitely Present) modifies “opacity” (OBS-DP), and “right and lower” (Anatomy) modifies “lobe” (Anatomy). “opacity” (OBS-DP) located at “lobe” (Anatomy). And “opacity” suggests of “infection” (OBS-DP). And “pneumothorax” (Observation Definitely Absent) is not presented in the X-ray scan. This shows an abstract of one patient’s reports content.

However, this is a confidential dataset, in order to show the result. We also apply the model on a public dataset called NCBI-disease [16] on huggingface. This dataset contains the disease name and concept annotations of the NCBI disease corpus, a collection of 793 PubMed abstracts fully annotated at the mention. It has 5433 training instances, 924 validation instances, and 941 test instances.

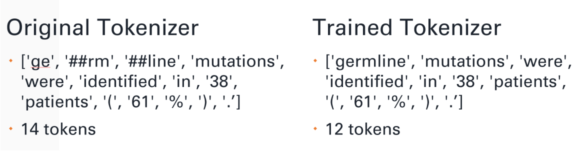
5 METHODOLOGY

For the NER task, in order to let the BERT know the medical words, we trained the tokenizer by sum up all text corpus in the dataset to a large corpus. But fit all text into memory is hard to achieve, we first separated the texts into multiple groups than used python generator to load the corpus as 1000 strings at a time. The process of training tokenizer is a statistical process. The result is based on the frequency of our dataset. After we have the trained tokenizer, we create our model from the original BERT-base checkpoint. And use the trained tokenizer to convert our input to word embedding tokens. Then we started to train our BERT-base model with and without a new tokenizer on our dataset. Once we have the best model of this task. We save the checkpoint to apply on a downstream task for examinations prediction. First. we retrieve the target label from the dataset as a sentence that contains all the examinations of one patient should do in her/his radiology reports. Then, we apply two different methods on this task. One is multi-class classification method. We found that in the over 9000 instances, the examinations only have 130 different combinations. Hence, we used those string sentence as our target label directly. For example, one sentence could be “XR Chest, Non Dedicated Unit-CH” as a examinations combination. The other method we used is multi-label classification. We use regular expression to retrieve each examination from the target label as separated tokens. Such as “XR Chest, Non Dedicated Unit-CH” to “XR Chest” and “Non Dedicated Unit-CH”. Each target label could have less than three examinations. We used one-hot encoding to represent them in numerical values.

6 Experiment Design, Model structures and Result

6.1. NER task

For the tokenizer training process, first we retrieve the original tokenizer from the BERT-Base model. Then we apply the training functions on the summary corpus directly. It took 5 minutes to finish the whole procedure. Then we apply the trained tokenizer on a medical domain sentence, such as “Germline mutations were identified in 38 patients 61%.”.



Original Tokenizer

- ['ge', '##rm', '##line', 'mutations', 'were', 'identified', 'in', '38', 'patients', '(', '61', '%', ')', '.']
- 14 tokens

Trained Tokenizer

- ['germline', 'mutations', 'were', 'identified', 'in', '38', 'patients', '(', '61', '%', ')', '.']
- 12 tokens

Figure 4. Tokenization demonstration

The trained tokenizer will recognize the medical word in one piece instead of a few sub-tokens.

By Figure 4, we can see that the trained tokenizer has the desired outcome in our methodology. However, in order to show the overall result. We sum up all the corpus in the dataset as one string. Then we used those two tokenizers to convert this string into multi-tokens.

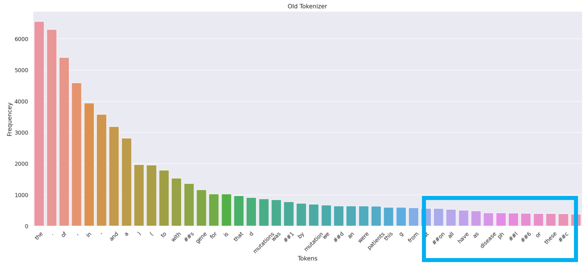


Figure 5. Original Tokenizer result on the whole dataset

By Figure 5, the original tokenizer result. We can see that in the blue square, there are a lot of cut out tokens in different ranks. And in Figure 6, the trained tokenizer, the result has more medical words in one token together.

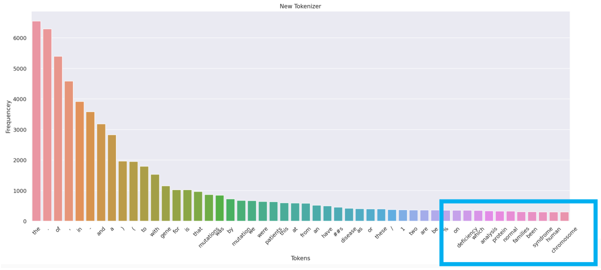
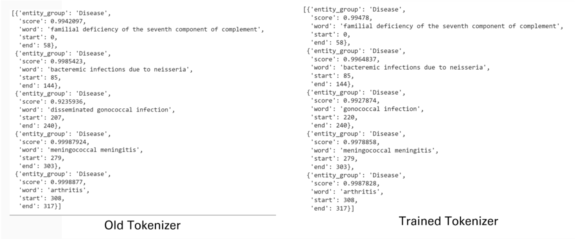


Figure 6. Trained Tokenizer result on the whole dataset

In order to compare the tokenizers performance, we apply the different tokenizers both on the BERT-Base model, with attention dropout rate 0.1, gelu activation function, 0.1 hidden layer dropout rate, 768 hidden units size, and max input tokens limitation with 512.

We take one reports as the inference example. “Familial deficiency of the seventh component of complement associated with recurrent bacteremic infections due to Neisseria . The serum of a 29 - year old woman with a recent episode of disseminated gonococcal infection and a history of meningococcal meningitis and arthritis as a child was found to lack serum hemolytic complement activity.” The model suppose to be extract the diseases from this sentence.



```
[{'entity_group': 'Disease',
  'score': 0.9942097,
  'word': 'familial deficiency of the seventh component of complement',
  'start': 0,
  'end': 58},
 {'entity_group': 'Disease',
  'score': 0.9985423,
  'word': 'bacteremic infections due to neisseria',
  'start': 85,
  'end': 144},
 {'entity_group': 'Disease',
  'score': 0.9235936,
  'word': 'disseminated gonococcal infection',
  'start': 207,
  'end': 240},
 {'entity_group': 'Disease',
  'score': 0.99987924,
  'word': 'meningococcal meningitis',
  'start': 279,
  'end': 303},
 {'entity_group': 'Disease',
  'score': 0.9998877,
  'word': 'arthritis',
  'start': 308,
  'end': 317}]
```

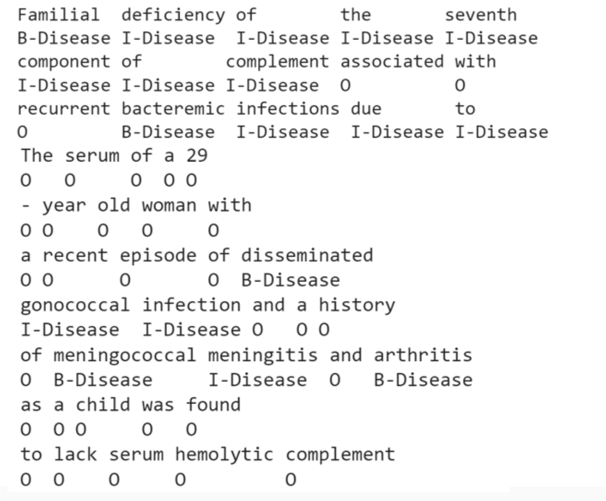
Old Tokenizer

```
[{'entity_group': 'Disease',
  'score': 0.99478,
  'word': 'familial deficiency of the seventh component of complement',
  'start': 0,
  'end': 58},
 {'entity_group': 'Disease',
  'score': 0.9964837,
  'word': 'bacteremic infections due to neisseria',
  'start': 85,
  'end': 144},
 {'entity_group': 'Disease',
  'score': 0.9927874,
  'word': 'gonococcal infection',
  'start': 220,
  'end': 240},
 {'entity_group': 'Disease',
  'score': 0.9978858,
  'word': 'meningococcal meningitis',
  'start': 279,
  'end': 303},
 {'entity_group': 'Disease',
  'score': 0.9987828,
  'word': 'arthritis',
  'start': 308,
  'end': 317}]
```

Trained Tokenizer

Figure 7. NER result

In Figure 7. The left part is the original tokenizer reports. We can see that for each retrieved token, it has 92% to 99% confidence. And the right side is the trained tokenizer performance, the model has 99% confidence on each prediction. However, if we look at Figure 8 as the ground truth. The trained tokenizer's result missed the word "disseminated" that should be the beginning token of "disseminated gonococcal infection".



```
Familial  deficiency of        the       seventh
B-Disease I-Disease  I-Disease I-Disease I-Disease
component of         complement associated with
I-Disease I-Disease I-Disease  O          O
recurrent bacteremic infections due       to
O         B-Disease  I-Disease  I-Disease I-Disease
The serum of a 29
O   O     O  O O
- year old woman with
O O    O   O     O
a recent episode of disseminated
O O      O       O  B-Disease
gonococcal infection and a history
I-Disease  I-Disease O   O O
of meningococcal meningitis and arthritis
O  B-Disease     I-Disease  O   B-Disease
as a child was found
O  O O     O   O
to lack serum hemolytic complement
O  O    O     O         O
```

Figure 8. NER ground Truth

Futhermore, if we observe the validation performance of those two models at Figure 9. We can see that the blue line as the original tokenizer model always has the better validation accuracy than the trained tokenizer. Even if after fine-tuning 20 epochs. We argue the reason is that the performance of BERT is based on how much similarities to its base model. Therefore, we decided to use the BERT base model to directly train our downstream task.

6.2. Examinations prediction

6.3. Multi-class Prediction

For examination prediction task, we apply our trained BERT-Base model directly on the same radiology dataset. First, we

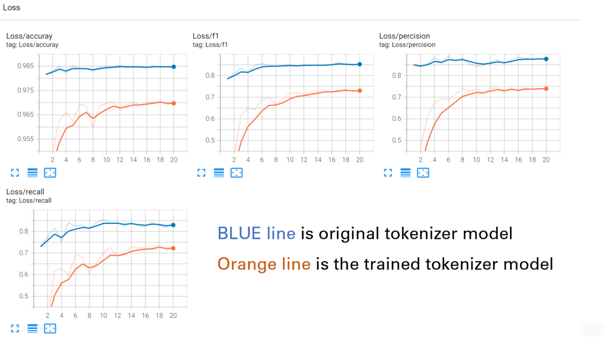
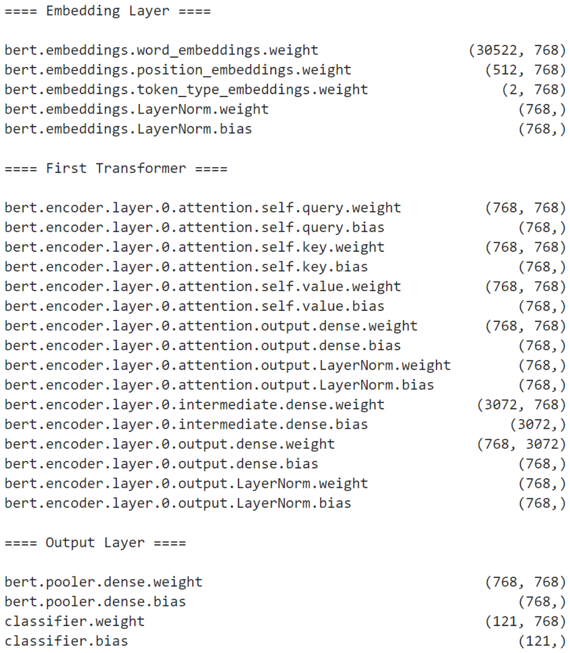


Figure 9. NER Validation Accuracy

retrieve the target labels as one examinations combination as one label. In total, we have 121 different labels. Such as "CT 3D Reconstruction -PE", "CT Abd, Pelvis, Combined -AB", "CT Abd, Pelvis, Non Enhanced -AB", "CT Abd,Pelvis, Enhanced -AB","CT Abdomen, Combined -AB", "CT Chest Pulmonary Embolus -CH", and etc. For the model structure, we set our model has 30522 total dictionary length, with 768 hidden units, and 121 units output as Figure 10 shows.



```
==== Embedding Layer ====

bert.embeddings.word_embeddings.weight                      (30522, 768)
bert.embeddings.position_embeddings.weight                    (512, 768)
bert.embeddings.token_type_embeddings.weight                    (2, 768)
bert.embeddings.LayerNorm.weight                                  (768,)
bert.embeddings.LayerNorm.bias                                    (768,)

==== First Transformer ====

bert.encoder.layer.0.attention.self.query.weight              (768, 768)
bert.encoder.layer.0.attention.self.query.bias                    (768,)
bert.encoder.layer.0.attention.self.key.weight                (768, 768)
bert.encoder.layer.0.attention.self.key.bias                      (768,)
bert.encoder.layer.0.attention.self.value.weight              (768, 768)
bert.encoder.layer.0.attention.self.value.bias                    (768,)
bert.encoder.layer.0.attention.output.dense.weight            (768, 768)
bert.encoder.layer.0.attention.output.dense.bias                  (768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight            (768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias              (768,)
bert.encoder.layer.0.intermediate.dense.weight               (3072, 768)
bert.encoder.layer.0.intermediate.dense.bias                     (3072,)
bert.encoder.layer.0.output.dense.weight                     (768, 3072)
bert.encoder.layer.0.output.dense.bias                            (768,)
bert.encoder.layer.0.output.LayerNorm.weight                      (768,)
bert.encoder.layer.0.output.LayerNorm.bias                        (768,)

==== Output Layer ====

bert.pooler.dense.weight                                      (768, 768)
bert.pooler.dense.bias                                            (768,)
classifier.weight                                             (121, 768)
classifier.bias                                                   (121,)
```

Figure 10. Multi-class model structure

We tried our model with batch size 32, 256 max tokens, from Figure 11, the model started to get overfitting after 5 epochs. The model has no significant improvement after fine-tuning on our checkpoint model after 20 epochs. And the model only only has 59% test accuracy. Therefore, we assumed it could cause by the 256 max tokens is limited and cut out the input sentences into a smaller rangle. Then, we trained the model on a batch size 16, with 512 max tokens setting. Since if we set the batch size as 32, the GPU memory will run out and we can not start the training process.

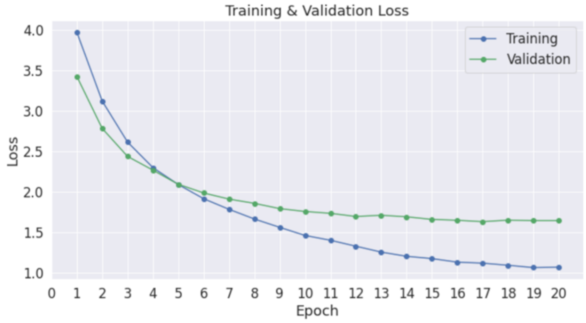


Figure 11. Train and Validation Loss

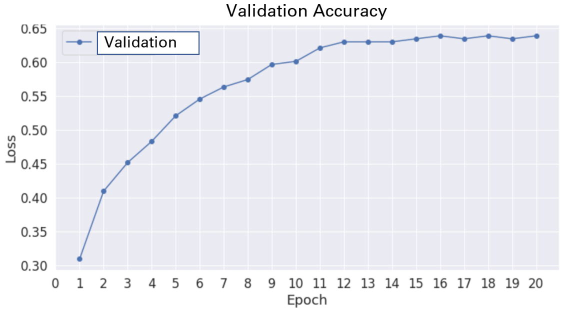


Figure 12. Validation Accuracy

For the batch size 16, 512 max tokens model, we have slightly better performance. It has lower loss in the training and validation loss with a 62% test accuracy. However, if we compare those two models, their validation and accuracy curve are similar. We argue that the different hyperparameter setting about max token slightly improve the performance, but can not improve it furthermore.

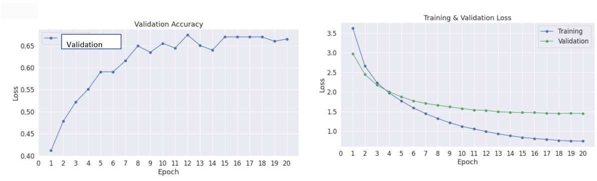
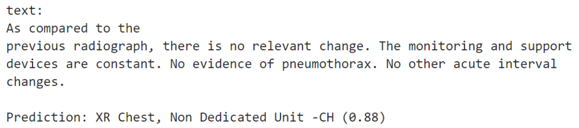


Figure 13. Batch size 16, max token 512

For the same example we talked in problem definition, “As compared to the previous radiograph, there is no relevant change. The monitoring and support devices are constant. No evidence of pneumothorax. No other acute interval changes.”, we use our batch size 16, 512 tokens model to inference its examinations types as Figure 14 shows.



```
text:
As compared to the
previous radiograph, there is no relevant change. The monitoring and support
devices are constant. No evidence of pneumothorax. No other acute interval
changes.

Prediction: XR Chest, Non Dedicated Unit -CH (0.88)
```

Figure 14. Inference result

We can see that the model has 0.88 confidence about the prediction “XR Chest, Non Dedicated Unit -CH (0.88)”. We also draw the confidence bar chart for observing the confidence level of all 121 labels as Figure 15 shows. We truncated the whole chart into small piece, since the other labels’ confidence level is almost 0. And the chart is too long.

From Figure 16, we can see that for the label “XR Chest, Non Dedicated Unit -CH”, it has 88% confidence, and nearly 12% for “XR Chest, Mobile 1 View”, and 10% for “XR Chest, 1 View -CH.”

We also draw an attention map for each word in order to see which one has the

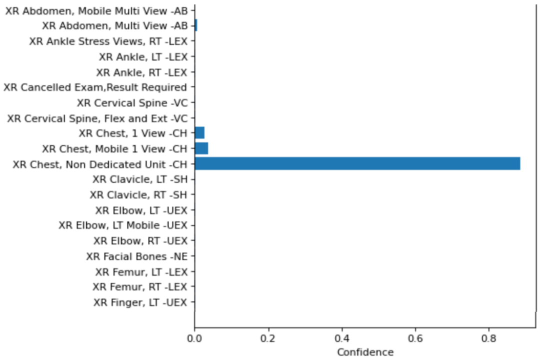
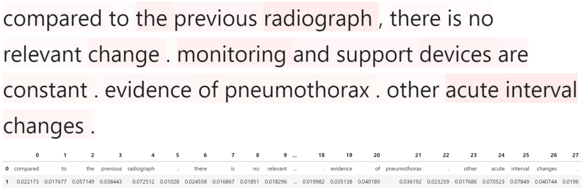


Figure 15. Confidence level bar chart

most influence on our prediction. The background color of each word shows its attention level. The more red it is, the higher attention it has.



compared to the previous radiograph , there is no relevant change . monitoring and support devices are constant . evidence of pneumothorax . other acute interval changes .

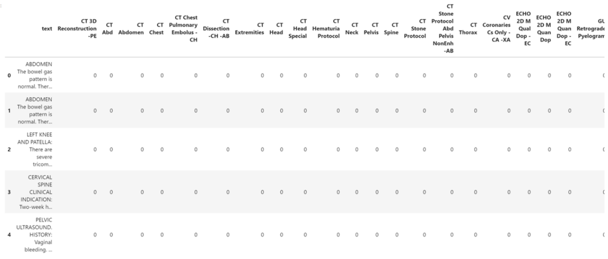
	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21	22	23	24	25	26	27
0	compared	to	the	previous	radiograph	,	there	is	no	relevant	...	.	evidence	of	pneumothorax	.	other	acute	interval	changes	.
1	0.022173	0.017677	0.057149	0.038443	0.072512	0.01028	0.024558	0.016867	0.01851	0.018296	...	0.019982	0.035138	0.040189	0.036192	0.023259	0.017686	0.070523	0.07849	0.040744	0.0196

Figure 16. Attention score of the input

We can see that the word “pneumothorax” and ‘acute interval changes” has the high attention score. And the word “pneumothorax” means ’A collapsed lung occurs when air escapes from the lung.’. That is the disease should be detected from CR Chest. Although we didn’t use the trained tokenizer, the BERT model still recognized this medical word by fine-tuning. It shows the BERT model has the ability to solve medical domain task, even if the original model has never seen those before. However, from its prediction, we can see that attention score related to different examinations. If we use the combinations as the target label, the accuracy may be lower than we use multi-label classification.

6.4. Multi-label prediction

In this method, we first use regular expression to retrieve all the examination from one single target label, then we use one-hot encoding to represent them in numerical values. At the end, we delete all duplicates rows to save memory space, as Figure 16 shows.



	text	CT 3D Reconstruction -PE	CT Abd	CT Abdomen	CT Chest	CT Chest Pulmonary Embolus - CH	CT Dissection -CH -AB	CT Extremities	CT Head	CT Head Special	CT Hematuria Protocol	CT Neck	CT Pelvis	CT Spine	CT Stone Protocol	CT Stone Protocol Abd Pelvis NonEnh -AB	CT Thorax	CV Coronaries Cs Only - CA -XA	ECHO 2D M Qual Dop - EC	ECHO 2D M Quan Dop	ECHO 2D M Quan Dop - EC	GU Retrograde Pyelogram
0	ABDOMEN The bowel gas pattern is normal. Ther...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	ABDOMEN The bowel gas pattern is normal. Ther...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	LEFT KNEE AND PATELLA: There are severe tricom...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	CERVICAL SPINE CLINICAL INDICATION: Two-week h...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	PELVIC ULTRASOUND. HISTORY: Vaginal bleeding. ...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 17. Target label demonstration

For training, we have batch size 8 with 256 tokens, with learning rate 1e-5. And F1 score as compute metrics for evaluating the model performance during the training. We can see that the F1 score and Roc Auc curve are relatively high at even the beginning. The accuracy started from 42% and it was slowly climbing to 62% in epoch 20.

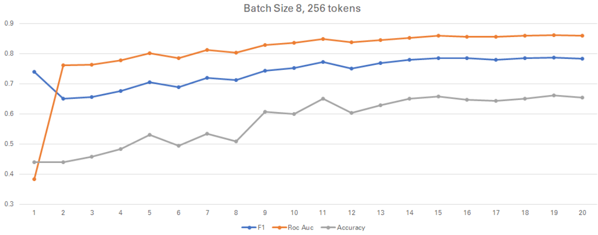


Figure 18. Multi-label Batch size 8, 256 max tokens

Hence, we trained another 140 epochs in total as 5 separated training. The accuracy started from 42% to 74%. The fine-tuning BERT model on this multi-label task works well. It has the ability to retrieve the most important tokens and gives them higher attention score as Figure 19 shows.

Table 1. Models Evaluation Result

Model setting	loss	f1	roc auc	accuracy	evaluation time
batch size 8, 256 max tokens	0.0248	0.8124	0.8972	0.7437	4.0027 secs
BERT-Base model	0.032235	0.6168	0.7397	0.4227	4.1021 secs

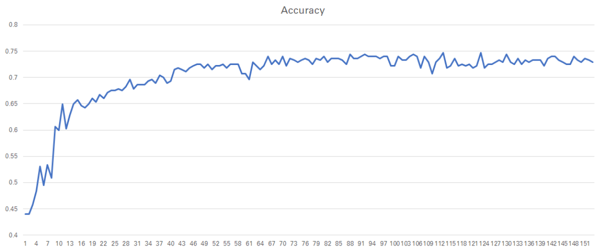


Figure 19. Multi-label Batch size 8, 256 max tokens

For the overall training, as Figure 20 shows. The model have a quick improvement at the first training, and the rest of training iterations have slowly improving but don't have overfitting problem. If we have more computation resources that can contain more than 256 max tokens, we may have even better result.

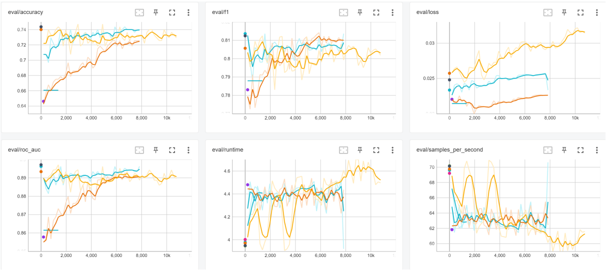
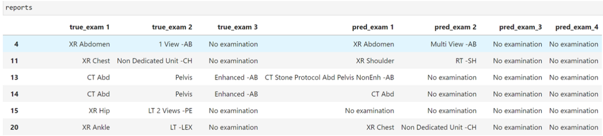


Figure 20. Multi-label Batch size 8, 256 max tokens

The overall test set evaluation has 74.37% accuracy, 0.0248 overall, 81.24% F1 score, 89.72% Ruc Auc score as Table 1 shows. The result is relatively satisfying. Based on the result of BERT-base model only has 42% accuracy.

For the inference, the left side is the true labels as Figure 21 shows. and the rest is the corresponding MODEL predicted examinations. And this table contains 6 wrongly classified instances. We can see that even the prediction is wrong. Some of the labels are correct. And the wrong predictions have similar meaning as the true label. Such as the first one, it should be 1 view AB, but the model thought it is multi view AB. That could be improvement in the future.



reports

	true_exam 1	true_exam 2	true_exam 3	pred_exam 1	pred_exam 2	pred_exam_3	pred_exam_4
4	XR Abdomen	1 View -AB	No examination	XR Abdomen	Multi View -AB	No examination	No examination
11	XR Chest	Non Dedicated Unit -CH	No examination	XR Shoulder	RT -SH	No examination	No examination
13	CT Abd	Pelvis	Enhanced -AB	CT Stone Protocol Abd Pelvis NonEnh -AB	No examination	No examination	No examination
14	CT Abd	Pelvis	Enhanced -AB	CT Abd	No examination	No examination	No examination
15	XR Hip	LT 2 Views -PE	No examination	No examination	No examination	No examination	No examination
20	XR Ankle	LT -LEX	No examination	XR Chest	Non Dedicated Unit -CH	No examination	No examination

Figure 21. Inference on the test set

7 CONCLUSION

By applying our RadiBERT on 2 different downstream tasks for training over more than 80 hours. We found that on certain medical domain corpus with fine-tuning on the BERT-base model is valid. And using a new tokenizer is not necessary if we have short input sentences or the fine-tuning BERT base model is already having good result. And if we have very long input tokens, we may use new tokenizer for reducing the length of tokenized output tokens. For the downstream task, large number of labels are hard to train and get valid result on most of models. But using fine-tuning on BERT, we can see that the model remembered the certain words/tokens should be more important than others. And by using the attention technique, the BERT does learn the medical words even without extra instructions. In the future, for certain downstream task, we can apply additional dictionary for further improving the model performance or adjust the word embedding structures before feeding into the model. And if we have more power GPU memory as the other researches have, we may save more tokens in the memory to achieve better result.

References

[1] Jain, S., Agrawal, A., Saporta, A., Truong, S. Q. H., Duong, D. N., Bui, T., … Rajpurkar, P. (2021). RadGraph: Extracting Clinical Entities and Relations from Radiology Reports. doi:10.48550/ARXIV.2106.14463

[2] Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. doi:10.48550/ARXIV.1606.05250

[3] Sang, E. F. T. K., De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. doi:10.48550/ARXIV.CS/0306050

[4] Rasmy, L., Xiang, Y., Xie, Z. et al. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. npj Digit. Med. 4, 86 (2021). https://doi.org/10.1038/s41746-021-00455-y

[5] Nakayama, H. (2018). seqeval: A Python framework for sequence labeling evaluation. https://github.com/chakki-works/seqeval.

[6] Choi, E. et al. RETAIN: An Interpretable Predictive Model for Healthcare using reverse Time Attention Mechanism. Adv. Neural Inf. Process. Syst. 29, 3504–3512(2016).

[7] Rajkomar, A. et al. Scalable and accurate deep learning with electronic health records. NPJ Digital Med. 1, 18 (2018).

[8] Baytas, I. M. et al. Patient Subtyping via Time-Aware LSTM Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 65–74 (ACM, 2017).

[9] Delbrouck, J.B., Chambon, P., Bluethgen, C., Tsai, E., Almusa, O., Langlotz, C.. (2022). Improving the Factual Correctness of Radiology Report Generation with Semantic Rewards.

[10] Alistair E. W. Johnson, Tom J. Pollard, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G. Mark, Seth J. Berkowitz, and Steven Horng. 2019. MIMIC-CXRJPG, a large publicly available database of labeled chest radiographs. arXiv e-prints, page arXiv:1901.07042.

[11] W. Yu . ., 'Dict-BERT: Enhancing Language Model Pre-training with Dictionary'. arXiv, 2021.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. arXiv, 2018.

[13] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, 'SpanBERT: Improving Pre-training by Representing and Predicting Spans'. arXiv, 2019.

[14] Y. Liu . ., 'RoBERTa: A Robustly Optimized BERT Pretraining Approach'. arXiv, 2019.

[15] Y. Gu . ., 'Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing', ACM Transactions on Computing for Healthcare, . 3, . 1, . 1–23, I 2022.

[16] R. I. Doğan, R. Leaman, and Z. Lu, 'NCBI disease corpus: a resource for disease name recognition and concept normalization', Journal of biomedical informatics, vol. 47, pp. 1–10, 2014.