Building a Natural Language Processing System to Characterize the Disease Progression in Radiology Reports

Stanford CS224N Custom Project

Gautham Raghupathi
Department of Computer Science
Stanford University
gautham@stanford.edu

Yusef Qazi
Department of Electrical Engineering
Stanford University
yqazi27@stanford.edu

Hamza El Boudali
Department of Computer Science
Stanford University
hamza410@stanford.edu

Abstract

NLP methods have been successfully used to extract clinical diagnoses from radiology reports; however, they haven't been explored as a way to incorporate information from previous radiology reports. In this paper, we seek to 1) summarize the change in disease in radiology reports to aid in the creation of future datasets with multiple timepoints for the Healthcare AI community and 2) implement BERT-based models that can predict disease progression at a higher accuracy than a rule based labeler. We find that BlueBERT outperforms regular BERT and Bio_ClinicalBERT when fine-tuned on manually annotated reports, but regular BERT outperforms BlueBert and Bio_ClinicalBERT when linear evaluation is used.

1 Key Information

- TA Mentor: Gaurab Banerjee

2 Introduction

Radiology reports contain important information about a patient's medical exams that can be used to train medical image classifiers. Chest X-rays are the most common radiological exam in clinical practice; however, chest X-ray radiology reports are often free-text and typically require medical domain knowledge to extract labels. Recently, NLP methods have been successfully used to extract clinical diagnoses from these reports; however, they haven't been explored as a way to extract disease progression. Disease progression is one of the most important aspects of a radiology report and therefore this problem is a natural next step in this trend of using NLP models to annotate radiology text reports for very important clinical applications.

In this paper, we propose a BERT-based model that is able to predict disease progression from single radiology text reports at a higher accuracy than existing methods. In order to train, finetune, and test this model, we also create a labeled chest X-ray dataset which will be useful for the Healthcare AI community. We do this by manually labeling examples and attempting to use backtranslation to augment the data. We also implement an automatic rule-based labeler to use as a baseline for comparison with the BERT-based model. In addition to this labeler, we test our BERT-based model against BlueBERT and Bio_ClinicalBERT.

We find that BlueBERT outperforms regular BERT and Bio_ClinicalBERT when fine-tuned on manually annotated reports, but regular BERT outperforms BlueBert and Bio_ClinicalBERT when linear evaluation is used. In both cases, the BERT-based models outperform the baseline rule-based labeler.

3 Related Work

Our work is similar to that of CheXbert [1], a state-of-the-art BERT-based model that extracts the presence of clinically important observations from free text radiology reports. Like our model, the CheXbert model is trained on the annotations of a rule-based labeler and fine-tuned on expert annotations augmented by backtranslation. Their task is different from ours however because we extract disease progression from a different section of the radiology text reports (*Findings*) than the section they use to extract their observations (*Impression*). We drew inspiration for our approach from this paper and our work is a natural next step in using these BERT-based methods for similar tasks.

Both our work and CheXbert use the CheXpert [2] rule-based labeler. This automated radiology report labeler is able to extract mentions of a list of key phrases from sections in radiology text reports. These phrases must be collected by human experts, and in the CheXbert and CheXpert papers, they use board-certified radiologists to curate the list of key phrases from radiology reports. The CheXpert labeler outperforms other radiology report labelers, such as the NIH labeler.

Another work in the same area as this paper and Chexbert's paper is RadGraph [3]. The authors of RadGraph extract clinical entities and relations from free-text radiology reports in order to use radiology reports for imporant healthcare applications. Unlike us and Chexbert, they are not extracting particular labels from specific sections of the reports in order to make clinical predictions. Rather, they develop an information extraction schema for structuring radiology reports, which they then use to annotate an inference dataset which they release. All of these papers annotate MIMIC-CXR reports, either exclusively or partially.

There are other machine learning methods used to detect change in chest X-rays that are different from ours, in that they do not rely on radiology reports. One such method is explored in *Longitudinal Change Detection on Chest X-rays Using Geometric Correlation Maps* [4], in which they use convolutional neural networks to extract feature maps for pairs of longitudinal chest X-ray images, in order to detect change in these images. They manually counted the frequencies of common sentences in each report and used these sentences in their model. They ignored the other hundreds of thousands of sentences, and left it to future work to employ natural language processing. We aim to overcome this limitation by taking advantage of state-of-the-art NLP methods.

4 Approach

4.1 Baseline: Rule-Based Labeling

Determining the disease progression from radiology report text can be accomplished using non-ML techniques. We will employ a rule-based labeler as our baseline to compare against our proposed NLP methods. Inspired by the CheXpert labeler [2], our rule-based labeler makes inferences for our 3-class classification problem ("Better", "No Change", "Worse") in a 3 stage process. In the first stage, mention extraction, the labeler searches for handpicked phrases in the free-text of the *Findings* section to associate with different disease progression classes. In the next stage, mention classification, we classify each of these associations as a negative, uncertain, or positive classification via dependency parsing. Specifically, we use regular expressions from the NegBio [5] library to identify grammatical negations and appropriately characterize each of our mentions. Finally, in the mention aggregation stage, we compile our class individual outputs from the mention classification stage into a single one-hot encoded vector. Each report will only have at most one positive classification and at least two negative classifications.

4.2 BERT Labeling

We propose to use the learned representations from a BERT encoder to facilitate the classification of disease progression in radiology text.

Given a pretrained BERT encoder, we acquire class specific information by extracting the first token of the encoder output (<CLS>). This token has an embedding size of 768 in all pretrained BERT architectures that we experiment with. We then attach a classification head (dropout and linear layer) to this token, to map the 768 intermediate outputs to 3 logits, each corresponding to a class. When training/inferencing text tokens through this model, we cap the length to 512 tokens (except for Bio_Clincal BERT for which we cap the length to 128 tokens) as is done frequently in other BERT implementations. Since many of our samples will have text that is considerably padded, we also generate an attention mask as an argument for the encoder. In this way, attention will only be applied to tokens that are meaningful for the task.

When training this model, we apply a standard cross entropy loss. Given our extreme class balance, we weight this loss to penalize incorrect calculations for classes with a smaller number of labels. Our weights are generated from the class distribution of our training set. We denote the training scheme outlined in the above two paragraphs above as fine-tuning of the model.

We assess the efficacy of a pretrained BERT encoder in a linear evaluation scheme. This scheme is the same as our fine-tuning scheme, except that we freeze all the layers of our BERT encoder so that the encoder representations are deterministic. Thus, only our classification layer is trained for the disease progression task.

4.3 Backtranslation

Backtranslation is form of data augmentation in which we take our inputs and translate it to an external language. Then, we translate from the external language back to the original language. Ideally, our original input and translated result would be the same, but as with any deep learning model, we do not achieve a deterministic result. However, we can use this variability in output to our advantage to train our model.

Essentially, we are able to attain alternate representations of radiology reports without having to label more data. These alternatives allow our Transformer model to learn more relationships between words in our training set since the translation will have similar, but not the same, syntactical structure. We expect this data augmentation approach to improve accuracy of our model. Additionally, we show that we are able to take radiology reports in other languages and use one-way translation to input reports into our main model.

For backtranslation, we use Neural Machine Translation (NMT) with a Transformer, building the system from scratch. Our model is language-agnostic, meaning we can use it with any source and target language. We set up two of these Transformers and use German as our intermediate language. The first Transformer translates radiology reports from English to German, and the second Transformer translates those reports back from German to English. We will take each report in our training set and generate a backtranslation, augmenting this new report into our training set with the same labels as before. We will then continue training our BERT models in the same fashion as before.

5 Experiments

5.1 Data

For our primary task of classifying disease progression in radiology reports, we are using the MIMIC-CXR dataset [6], particularly, the free-text radiology reports. We use canonical Python preprocessing techniques to extract the *Findings* section from each of the radiology text reports and compile them in a CSV file for our DataLoader during the labeling and deep-learning tasks.

Since we are implementing a novel classification system, there are no existing labels for our free-text data. Therefore, we manually labeled the *Findings* sections of reports from the MIMIC-CXR dataset. For our project, this involved our team reading through reports and extracting key words and phrases that indicated "Better", "No Change", or "Worse" in disease progres-

Split	Class		
	"Better"	"No Change"	"Worse"
Train	0.087	0.553	0.360
Validation	0.100	0.500	0.400
Test	0.067	0.500	0.433

Table 1: **Class Distribution Across Train, Validation, and Test Sets**

sion. We labeled the positive class as a "1" and the others as "0", resulting in a one-hot encoded vector.

Ideally, we would like to have received proper labels from a certified radiologists, but time constraints did not allow for that to transpire. We are still confident that our labels are highly accurate and can be used to train a model. In total, we labeled 334 radiology reports amongst the 3 classes. With a proper rule-based system and backtranslation system, we can create more training labels without any additional manual work.

Our BERT-based classifiers use these 334 samples and splits them into training, validation, and test sets. The training set is 150 samples, the validation set is 50 samples, and the test set is 134 samples. As shown in Table 1, our data is highly unbalanced, with the "Better" class comprising of 10% or less of each split. The reports are stored as strings and are tokenized to fit the specific pretrained BERT encoder that we use (BlueBERT [7], Bio_ClinicalBERT [8], and BERT [9]).

For our secondary task of implementing a backtranslation system, we are using the Multi30K [10] dataset. This includes many image and text data points, but for the purposes of our task we will be using the German-to-English and English-to-German sentence pairs. To prepare the data, we are preemptively creating German and English vocabularies from the Spacy library and using their tokenizer to create the proper embeddings needed to input into our NMT Transformer. The output of the individual translation systems is a vector of tokens that contains potential "<unk>" tokens and padding tokens. Before sending this output into the second Transformer for backtranslation, we will untokenize and retokenize to ensure we get the proper start, end, and padding tokens.

5.2 Evaluation method

We will be using F1 scores to assess the performance of our BERT-based systems. We will also be using F1 scores to measure the performance of our rules-based labeler, to compare improvements between the baseline and deep learning solutions.

We will also be computing BLEU score to assess the effectiveness of our Neural Machine Translation system. Specifically, we will compute these scores for English-to-German and German-to-English separately. We would want our BLEU score to be relatively high to ensure that our reports retain their syntax, semantics, and context through the double translation process.

5.3 Experimental details

5.3.1 BERT-Based System

For our BERT-based system we seed all libraries and turn the determinstic flag on in PyTorch for the reproducibility of our results. We train over 20 epochs, with a patience of 3 epochs. We save the checkpoint with the lowest validation cross entropy loss. We use a batch size of 32 and a learning rate of 0.02. Each epoch took approximately 2 minutes to train on a Tesla V100 GPU with 16GB of VRAM (courtesy of Microsoft Azure).

5.3.2 Backtranslation

For the backtranslation Transformer, we are using a PyTorch Transformer with our own definition of word and positional embeddings for both the source and target languages. We use these embedding layers in the forward step of training our model to create attention masks for inputs of different lengths.

We are training each backtranslation model for 20 epochs with a learning rate of $3 * 10^{-4}$ and a batch size of 32 samples. Our embedding size is 512 and we are using 3 encoding and decoding layers. We are also using a dropout layer with a probability of 0.1 to prevent overfitting from our training data.

Training and BLEU score computation for each direction of translation took around 50 minutes on a Tesla V100 GPU with 16GB of VRAM.

5.4 Results

We computed F1 scores for our different models in both finetune and linear evaluation configurations. We found that when finetuning the entire BERT-based model, BlueBERT performs the best with an F1 score of **0.536**. This is what we expected considering the BlueBERT is pretrained on clinical notes from PubMed and likely has favorable parameter weights for our given task. However this performance is only 0.004 greater than BERT. We conclude that this difference is statistically insignificant, given that the p-value for this increase is much greater than 0.05 since we only trained on 200 samples. In linear evaluation, we see that the original BERT model barely outperforms the others, but this result is also statistically insignificant. We do see that our BERT-based models outperform the baseline results from the CheXpert labeler.

We note the peculiarity of our linear evaluation results with respect to our finetune results. We expected finetune results to exceed those of linear evaluation for each of the pretrained encoders. However, we predict that finetune underperformed linear evaluation due to the lack of our data. Our pretrained encoders each had over 108 million parameters while our classification head only consisted of 2.3 thousand parameters. With just 200 training samples, we could not have reached SOTA performance for our finetune training scheme (108 million parameters). However, since all our pretrained models are at equal disadvantage with respect to this regard, we can still compare their finetune results.

BERT Pretrained Encoder	Configuration	
	Finetune	Linear Evaluation
BlueBERT	**0.536**	0.672
BERT (Regular)	0.532	**0.679**
Bio_ClinicalBERT	0.527	0.657
Baseline	0.474	—-

Table 2: **Finetune and Linear Evaluation F1 Metrics for Different Pretraining Methods**

For backtranslation, we did not have any baselines simply due to the nature of Neural Machine Translation. Our BLEU score for German-to-English translation was **32.99**. Our BLEU score for English-to-German translation was **31.67**. Both of these BLEU scores were computed on the test set for Multi30K, which contained 1000 samples. We show the validation loss for German-to-English in Figure 1 and the validation loss for English-to-German in Figure 2.

6 Analysis

For the BERT-based model we find that BlueBERT is our best finetune performer (0.536) and BERT (regular) is our best linear evaluation performer (0.679). All BERT-based models outperform our baseline in both finetune and linear evaluation.

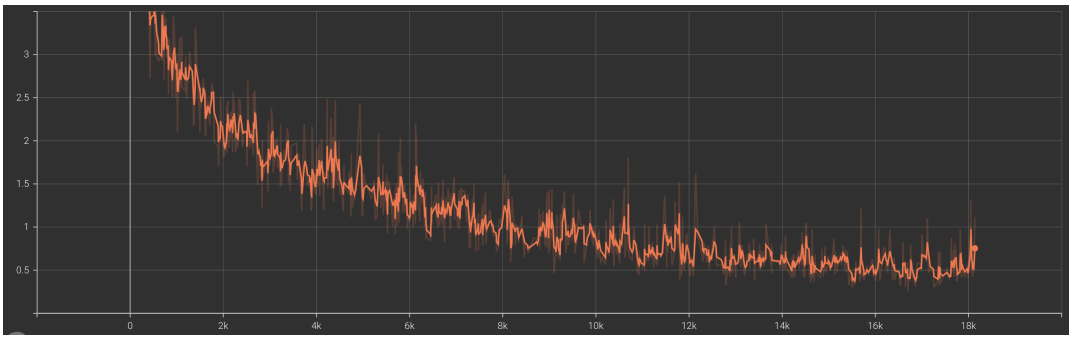

Figure 1: **Validation Loss for German to English**

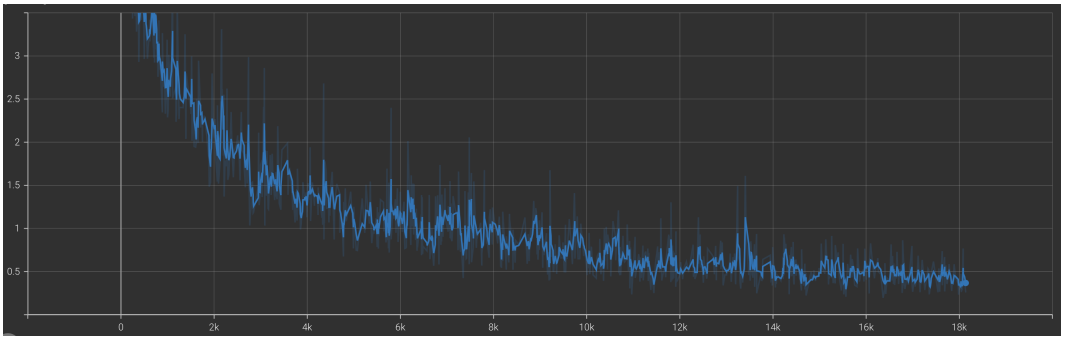

Figure 2: **Validation Loss for English to German**

We propose that backtranslation is a key enhancer to the BERT-based models' overall learning. Here, we will analyze the effect of backtranslation on some of the radiology reports from our curated dataset in Table 3. We see that the backtranslation outputs appear to be quite nonsensical. This portrays a key limitation in our project that is particularly due to the nature of the data that we are handling. The Multi30K dataset and the Spacy vocabulary sets for English and German contains words and sentences that are used in everyday language. The data that we are dealing with in this experiment contains heavy medical jargon and does not necessarily have German counterpart words that are common enough for the vocabulary sets to include them. Thus, during the first translation from English to German, many of the output tokens are <unk> tokens and the original words are lost. In the second translation from German to English, these <unk> tokens dominate the input and thus a proper English backtranslation cannot be achieved.

We know that our backtranslation implementation works due to the relatively high BLEU scores that we achieved after training our NMT models for 20 epochs individually. We presume that a more curated backtranslation dataset, one that contains medical observations and/or notes in another language, would provide better results.

7 Conclusion

We explore the effect of various BERT pretraining strategies on the performance of characterizing disease progression in radiology free-text reports. We find that the use of a domain-specific pretrained BERT encoder, namely BlueBERT, has improved performance over the traditionally pretrained BERT encoder. Furthermore, BERT-based models outperform rule-based labelers, as expected.

To the best of our knowledge, analysis of radiology reports has gone so far as to make inferences at a single time point (time at which the report is recorded). However, clinicians typically consult a patient's clinical history when making medical decisions. Creating a dataset that characterizes the disease progression from prior data points is the first step in allowing deep learning methodologies to incorporate multiple data points to come to medical decisions. Such deep learning methodologies are

Findings	Backtranslations
the cardiomediastinal silhouette is normal. there is no pleural effusion or pneumothorax. there is no focal lung consolidation. views of the upper abdomen are normal.	church members <unk> <unk> , <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> .
a pigtail catheter now overlies the left pleural space. there has been interval reexpansion of the left lung with a now small left pneumothorax, significantly decreased from comparison study. right basilar opacity may be due to atelectasis. superiorly the lungs are clear. hilar structures and cardiomediastinal silhouette is normal.	fast <unk> , washington <unk> <unk> to <unk> the <unk> <unk> to stop the <unk> <unk> <unk> who <unk> <unk> .
the heart is mildly enlarged. the mediastinal and hilar contours appear unchanged. there is no definite pleural effusion or pneumothorax. the lungs appear clear. mild degenerative changes are similar along the thoracic spine. there has been no significant change.	the <unk> <unk> , <unk> <unk> bank <unk> <unk> <unk> <unk> <unk> the <unk> <unk> against the <unk> football goal .

Table 3: **Outputs of Backtranslation on Sample Report Findings**

likely to outperform the current single time point methodologies since they have the capability to be more personalized for each patient. Furthermore, clinicians will can have greater confidence in these recommendations, since they factor more clinically relevant information.

An important future step for this project would be to acquire translation data for the medical domain. Similar to how BlueBERT and Bio_ClinicalBERT are specialized for text in the medical domain, it would be more appropriate to train a backtranslation system with task-specific data. Overall on the data front, acquiring more labeled data and vetting it with cardiologists would improve the performances of our methods.

References

[1] Akshay Smit, Saahil Jain, and Pranav Rajpurkar. CheXbert: Combining Automatic Labelers and Expert Annotations for Accurate Radiology Report Labeling Using BERT. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[2] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. In *Association for the Advancement of Artificial Intelligence*, 2019.

[3] Saahil Jain, Ashwin Agrawal, and Adriel Saporta. RadGraph: Extracting Clinical Entities and Relations from Radiology Reports. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[4] Dong Yul Oh, Jihang Kim, and Kyong Joon Lee. Longitudinal change detection on chest x-rays using geometric correlation maps. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 748–756, Cham, 2019. Springer International Publishing.

[5] Yifan Peng, Xiaosong Wang, Le Lu, Mohammadhadi Bagheri, Ronald Summers, and Zhiyong Lu. Negbio: A High-Performance Tool for Negation and Uncertainty Detection in Radiology Reports. In *American Medical Informatics Association (AMIA)*, 2017.

[6] Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih ying Deng, Roger G. Mark, and Steven Horng. Mimic-cxr, A De-Identified Publicly Available Database of Chest Radiographs with Free-Text Reports. In *Scientific Data 6*, 2019.

[7] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

[8] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[10] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics, 2016.