ATTENTION BASED MODELS ON HEALTH INFORMATICS DATA

Final Report

Information Technology/Data Science Capstone Project

COMP5703/DATA5703

Group Members

1. Kanishka Mohaia (480559773)
2. Kasidej Cha-umpong (500336122)
3. Yu Hin Elroy So (309004314)
4. Utkarsh Bele (490590751)
5. Siddhan Khondge (480557540)

Abstract

In recent years there has been unprecedented growth in the amount of biomedical research coupled with advances in deep learning and NLP space has invigorated interest in applying attention based models [4] to the biomedical domain. This research project takes inspiration from BioALBERT (Naseem et al, 2020) [10] where the BioALBERT models have been further extended and fine-tuned with new biomedical datasets on new downstream tasks namely named entity recognition, relationship extraction, sentence similarity, document classification and inference.

In addition, four new BioALBERT models have been pre-trained with clinical notes data from the MIMIC-III database and fine-tuned on the same downstream tasks mentioned earlier. The models are listed below.

- BioALBERT 2.0 (+ PubMed + MIMIC III 200K) [Base]
- BioALBERT 2.0 (+ PubMed + PMC + MIMIC III 200K) [Base]
- BioALBERT 2.1 (+ PubMed + MIMIC III 270K) [Large]
- BioALBERT 2.1 (+ PubMed + PMC + MIMIC III 270K) [Large]

Finally the suitability of using training attention based models on vaccine tweets has also been investigated in this project where BERT was pre-trained on vaccine related tweets and fine-tuned with sentiment analysis tasks and compared to CT-BERT as benchmark however this experiment was impeded by a bug in the tensorflow 2.1 and the team wasn’t able to continue with this experiment..

This research project was able to produce models that were able to outperform SOTA models in some tasks, it also managed to improve BioALBERT 1.0 and 1.1 by extending it with extra tasks that outperformed SOTA models.

Table of Content

1. Introduction

There has been a surge of activity in biomedical text mining using natural language processing techniques mainly due to greater advances in the natural language processing techniques such as attention based techniques [4].

Current research is mainly focused on extracting this information from sources such as pubmed and pmc [6], where millions of research articles are published. These articles are a great source of data for machine learning researchers and great work has been done on training BERT and ALBERT on domain specific corpus and it has shown that these domain specific models have performed well on these tasks when compared to benchmarks that were trained on general corpus.

This is mainly because biomedical research papers use quite a lot of domain specific jargon that may not be found in general corpus. However these models such as BioBERT[6], BLUEBERT[7], and BioALBERT[10] have been trained on very sanitised data as abstracts of research papers are mostly well written with the correct grammar and correct spellings. It is unlikely that five different papers might spell the same word in five different ways however this is a reality that can happen in a real world clinical setting where five different doctors might spell the same medicine in slightly different ways posing a greater challenge to NER tasks to extract the name of the medicine from a sentence.

This research project aims to answer the following questions.

1. How do BioALBERT and BioBERT perform when trained on downstream tasks created from clinical datasets?
2. Can an extended BioALBERT model trained on clinical notes data from real patients (MIMIC-III) outperform the current BioBERT and BioALBERT?

Hence BioALBERT was trained on downstream tasks namely extraction, named entity recognition, sentence similarity, document classification and inferencing with data from a variety of sources that has been widely used in the Biomedical NLP community as shared tasks. BioBERT was not trained on these tasks as the BLUE benchmark already includes results of experiments on BioBERT trained on clinical tasks.

In addition BioALBERT has been pre-trained on the MIMIC-III dataset which is a large database consisting of clinical related data associated with more than 40000 patients producing four models namely which have been further trained on the above mentioned downstream tasks.

- BioALBERT 2.0 (+ PubMed + MIMIC III 200K) [Base]
- BioALBERT 2.0 (+ PubMed + PMC + MIMIC III 200K) [Base]
- BioALBERT 2.1 (+ PubMed + MIMIC III 270K) [Large]
- BioALBERT 2.1 (+ PubMed + PMC + MIMIC III 270K) [Large]

This project also aimed to train attention based models on vaccine related tweets in addition to the MIMIC III datasets however due to a bug in Tensorflow 2.1 this investigation didn't produce desirable outcomes and will be discussed in the following sections in detail.

2. Literature review

Recent research activity in deep learning, attention based models as well as biomedical text mining have provided a strong foundation for other researchers to build upon this work and be able to come up with their own breakthrough research. This project heavily relies on transfer learning techniques, attention based models, attention and transformer models and specifically BERT, ALBERT, BioBERT, BioALBERT and CT-BERT are the main papers reviewed. The following sections provide a very detailed review of the techniques.

2.1. Transfer Learning

Transfer learning is a machine learning technique of adding task-specific to the last few layers of a large pre-trained model, then fine-tune to obtain the results. It reused the old model as the starting point for a new model. Transfer learning is suited for NLP specific task reason being:

1. Save time

The BERT model has already been trained on a lot of information about language. We only have to apply its weights to a new task but use it as initialization before fine-tuning. Moreover, training BERT from scratch is not practical as it takes hundreds of GPU hours.

2. Use less data

Since fine-tuning weight was initialized with the pre-trained weight from the original BERT model, this allows us to accomplish fine-tune with much smaller size data than training the model from scratch. This is the drawback for many NLP models, it requires a lot of data to train the network to achieve decent results which causes researchers viable time and money.

3. Good Results

From experiments conducted on BERT, It has shown that the simple fine-tuning procedure of BERT was able to achieve state of the art performance on multiple NLP tasks: inference, semantic similarity, classification, name entity recognition, etc.

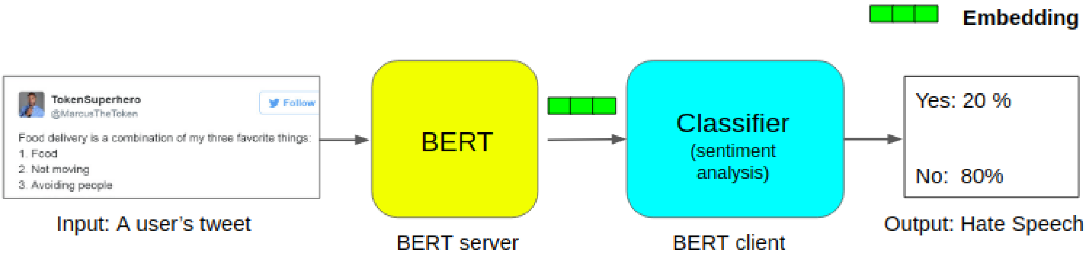
2.1.1. Pre-training

Simply put, pre-training in machine learning is to train a model on one task to create parameters that can be applied to other similar tasks. Therefore the model does not have to be trained again from scratch. It is able to transfer the previous weight and initialize it on new tasks. The parameters from the original pre-trained model will improve the new model's performance.

In the case of BERT, the researchers trained BERT on MLM(Masked Language Model) and NSP(Next Sentence Prediction) tasks on massive text dataset. The running cost them tens to hundreds thousand dollars of TPUs compute time, then they made the trained model available to the public for everyone to use.

2.1.2. Fine-tuning

Fine-tuning is the process of using pre-trained weights of a model as initialization for a new model being trained on different sets of data. It can be used to speed up the training process. In the case of BERT, a layer of untrained classifiers is added to the last layer of the pre-trained model and train it on a new dataset of a particular text classification task. Figure below shows an example of applying the pre-trained BERT to sentiment analysis by making changes to the last layer.



TokenSuperhero
@MarcusTheToken

Food delivery is a combination of my three favorite things:
1. Food
2. Not moving
3. Avoiding people

Figure 1: Applying BERT to text classification task [43]

2.2. Language models

2.2.1. Attention

Sequence-to-sequence models are popular deep learning models that achieve outstanding performance in NLP tasks. However, one major problem with these models is that they are unable to deal with long sentences due to diminishing effect on RNN. An improvement has to be made.

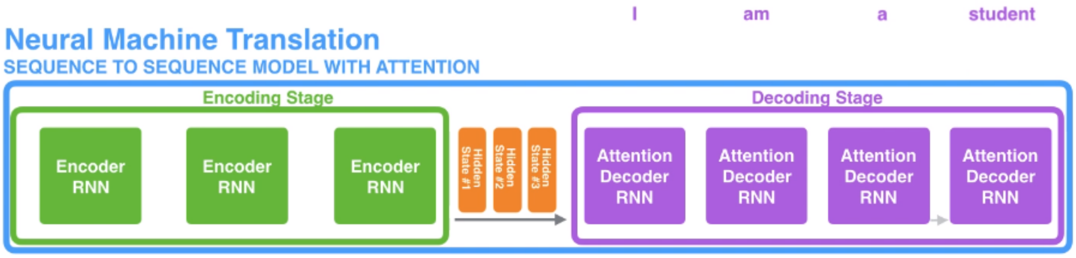


Figure 2: example of Seq2Seq model with attention [27]

Attention was introduced to counter this problem. It allows the model to focus on the important parts of the input sequence as a result the encoder can pass more data to the decoder[3]. It does this by instead of only passing the last hidden state of the encoding stage, the encoder with attention will pass all the hidden states to the decoder. The decoder looks at the set of hidden states it received from the encoder and gives each hidden state a weight. Then it multiplies each hidden state by its soft maxed weight. Doing so, it will reduce the effect of hidden states with low weight and amplify the hidden states with high weight. Figure below shows which part of the input word the model pays attention to at each decoding step.

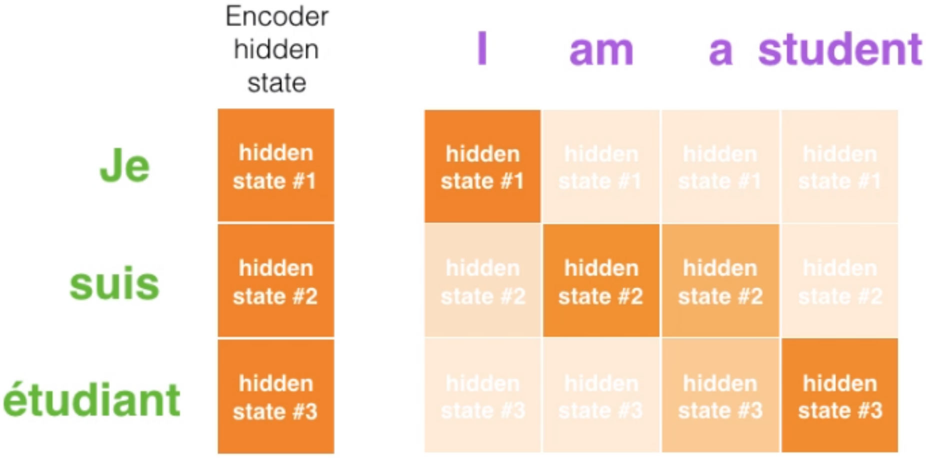


Figure 3: Attention at each decoding [27]

In Attention Is All You Need paper, a new type of attention was introduced, Multi-Head Attention. The original attention is a single linear attention function. Multi-head attention runs several single attention layers in parallel and their outputs are concatenated then multiplied by a matrix(jointly learnt matrix) to bring down the dimension back as though it was a single attention[4]. By concatenating attentions, it allows the model to attend to information from different subspaces at different positions. Multi-head attention significantly reduces the domination of attention by a single word and avoids a word attending too much on itself .

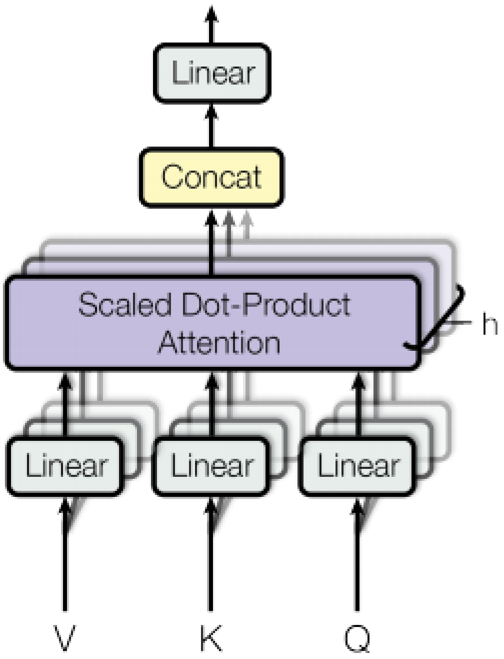


Figure 4: Multi-Head Attention [4]

2.2.2. Transformer

Transformer is a model that utilizes the advantage of attention to boost the training speed. Figure below shows an overview of the architecture of the transformer.

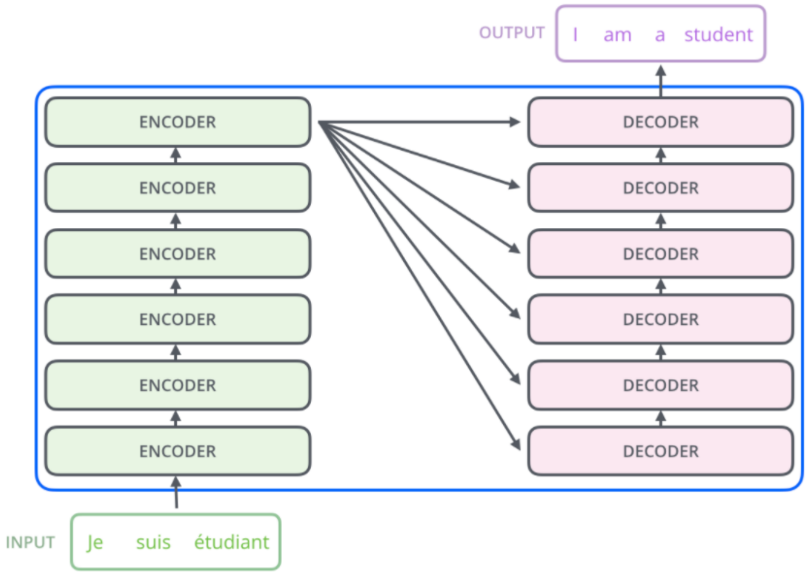


Figure 5: overview of transformer architecture [27]

Before sending the input to the first encoder(bottom most encoder), the inputs have to be embedded to change words into vectors. The encoder receives this list of vectors as input. It processes these vectors by passing them into a self-attention layer, then feed-forward neural network, then send the output out of the encoder upwards to the next encoder until the top encoder is reached. Self-attention layer allows the encoder to look at other positions in the input sequence which helps improve encoding. The top encoder transforms its output into a set of attention vectors. These vectors are later used by the encoder-decoder attention layer which allows the decoder to focus on appropriate sections in the input sequence. One important thing to note about the transformer is that each input flows through its own path in the encoder and these paths can be executed in parallel[4].

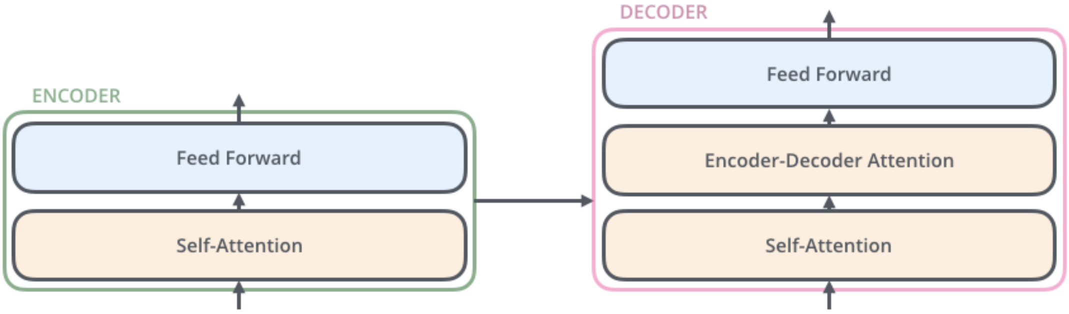


Figure 6: Encoder and Decoder [27]

2.2.3. BERT

BERT(Bidirectional Encoder Representations from Transformers) was originally introduced at the end of 2018 and has become a popular topic of NLP research ever since. This is mainly due to it being able to achieve outstanding understanding of languages and obtaining results as good as humans on certain tasks.

Before the release of BERT, most of the language representation models were designed either as Transformer based models trained unidirectionally like OpenAI GPT or LSTM based models from both sides like ELMo[7]. Unlike the model previously mentioned, BERT was designed to overcome the major limitation that standard Transformer based language models have, which is unidirectional training only from left to right. BERT introduced transformer encoders to make embedding tokens connect to each other using multi-attention heads.

To allow BERT developing language understanding during pre-training, the BERT development team introduced two unsupervised tasks, Masked Language Model (MLM) and Next Sentence Prediction (NSP) as pre-training tasks. To implement the Masked Language Model task, BERT randomly masks roughly 15% of the input and sometimes randomly replaces some words to others[1]. The objective of this task is to let the model predict the masked token based on both left and right context. This task allows BERT to pretrain a bidirectional Transformer on text without any labelling. Therefore, BERT can be pre trained on any type of corpus. The second task, Next Sentence Prediction (NSP), requires the model to predict whether two sentences are consecutive to each other. It gives BERT the ability to handle NLP tasks with text pair input. The original BERT model was pertained using BookCorpus (800M words) and text passages from English Wikipedia (2500M words).

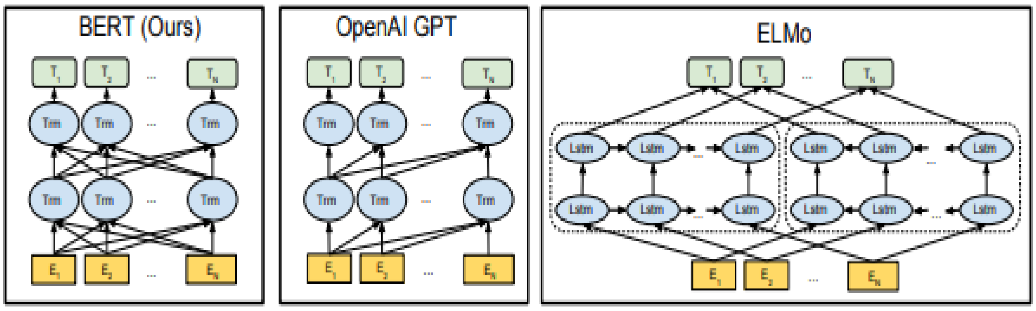


Figure 7: Comparison of model architecture [1]

BERT adapts to deal with various types of downstream tasks by using transfer learning and fine tune a pretrained model on new datasets with labels. To fine tune BERT for a specific task, for instance, BERT first takes vectors and weight matrices from a pretrained model by transfer learning and adds a new bottom layer to the end of the model. It then uses supervised learning to fine-tune the parameters of the final classifier with specific outputs[1].

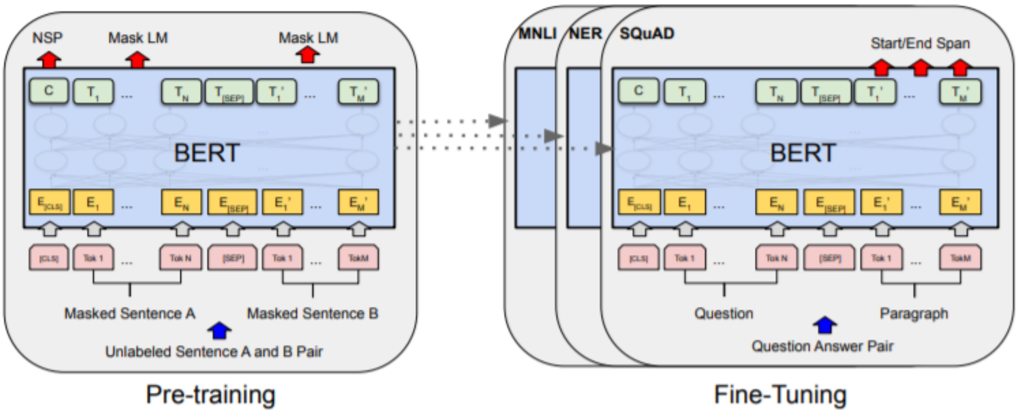


Figure 8: Procedure of pretraining and fine tuning on BERT [1]

BERT is able to process any input text with its own built-in BERT tokenizer. However, BERT performance may be tolerated if the input is domain specific and contains words not in the pre-training such as biomedical domain. The tokenizer that BERT uses is called WordPiece. This tokenizer contains 30000 token vocabulary and special tokens. If a word is not included in WordPiece vocabulary, then it will break the word into subwords[12]. If it cannot be splitted to subwords, it will break down further into individual characters. After that, word tokens will be converted to corresponding embedding id sequences as input.

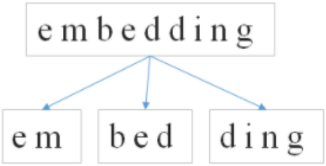


Figure 9: Wordpiece tokenizer break word into subword [12]

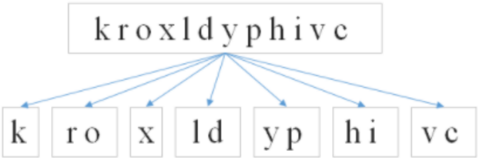


Figure 10: Wordpiece tokenizer break word into tokens [12]

Beside Token embedding, special tokens [CLS] and [SEP] are inserted to the start and the end of the input sequence to recognize a complete input sentence. If there are two sentences in the input sequence, another [SEP] token is added to split the sentences. Segment embedding, which is used to distinguish the segment of a token, and position embedding, which shows the position of a token, are also added to the word token embedding to get input embedding[1].

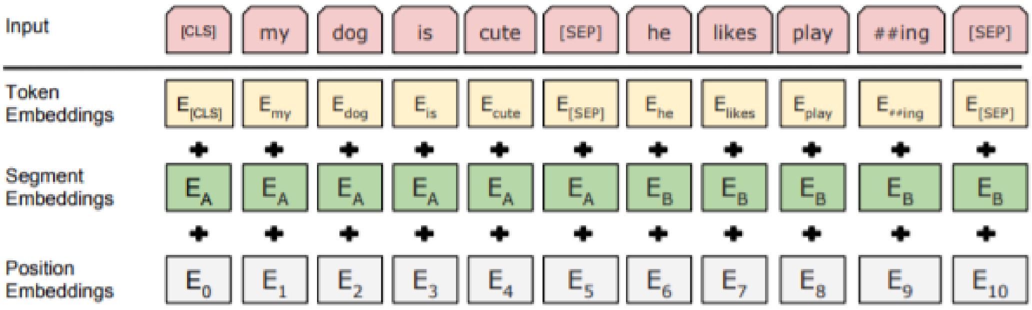


Figure 11: Procedure of embedding [1]

The BERT-LARGE and BERT-BASE models are two models released by the team with different sizes in layers, hidden units and number of attention heads. BERT-BASE has architecture (Layer=12, Hidden Size=768, Attention head=12, Total Parameters = 110M) and BERT-LARGE has architecture (Layer=24, Hidden Size=1024, Attention head=16, Total Parameters=340M)[1]. The graph below illustrates the interaction of each embedding to the layers. Input sequence will go through a transformer layer and output another sequence as the input with the same length for the next layer. Each word is incorporated to other words in the sequences. BERT has 12 layers with identical structure but different weight metric on each layer for the BERT-BASE model and 24 layers for the BERT-LARGE model.

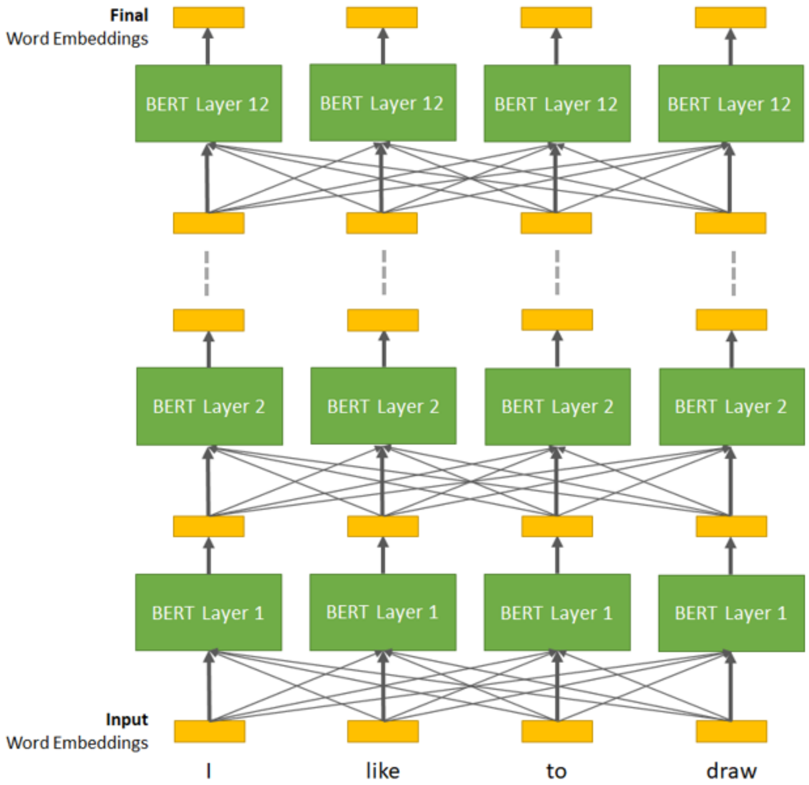
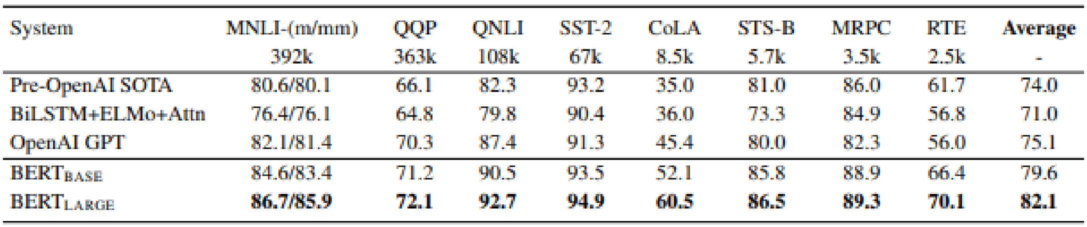


Figure 12: Layer structure inside BERT [12]

According to the experiment conducted by the BERT development team, BERT-LARGE significantly outperforms BERT-BASE across all NLP tasks in GLUE benchmark. Although BERT has good performance on tasks with general corpas, its performance on domain specific tasks still needs to be improved. Large scale of parameters also limits its pre-training speed, hence restricts the number of training steps.



System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT$_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT$_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results[1]

2.2.4. ALBERT

AlBERT, which stands for A Lite BERT model, is a modified version of the BERT model. Improved performance on the network with larger size on BERT shows the importance of network size on achieving SOTA performance[2]. Intuitively, developing a BERT model with more parameters can lead to better performance. The biggest issue on BERT is the memory limitation of hardware. BERT contains hundreds of millions parameters. The runtime and memory consumption would be largely increased by adding more layers on BERT. Effective use of large scales of parameters can enhance model performance. Based on this idea, ALBERT is designed to address the problem of GPU/TPU memory bottleneck and lack of efficiency on training BERT.

ALBERT has a similar Transformer based layer structure to BERT, however, the development team made some significant changes on BERT architecture. The main idea behind ALBERT is to preserve most of the capability of BERT with largely reduced parameters[13]. By achieving this goal, ALBERT can be trained on much more steps compared to BERT using the same time pand hence improve the performance. They introduced two parameter reduction techniques to reduce memory consumption, Parameter sharing and Factorized embedding parameterization.

In BERT, parameters across different layers are distinct. The model thus needs to update parameters on each layer during pre-training. ALBERT reduces the distinct parameters by using cross layer parameter sharing. This technique allows each layer to share the same set of parameters, which means that unique parameters would not grow with the increase in the number of layers[13]. Although parameter sharing slightly decreases the performance of the model, a deeper model can be built with much less GPU memory consumption and solve the GPU/TPU memory bottleneck problem. However, the model still needs to go through the parameters on each layer during training, that makes the time spent on the training process same as BERT.

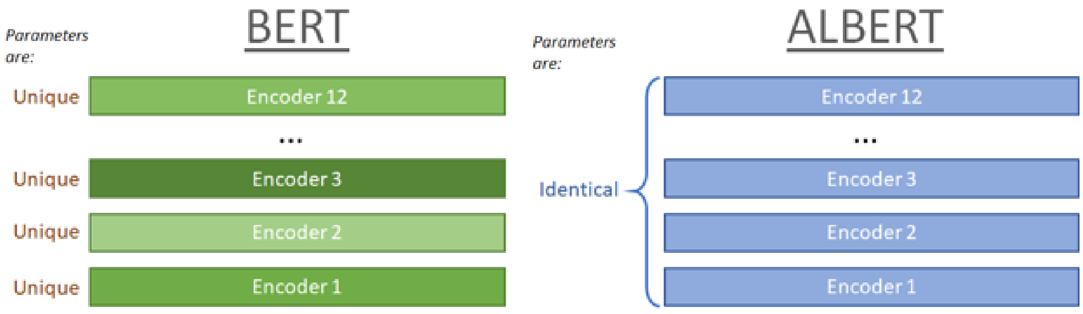


Figure 13:Layers of BERT and ALBERT [2]

The second parameter technique ALBERT used is Factorized embedding parameterization. In BERT, the WordPiece embedding size E is the same as the hidden layer size H. The ALBERT development team found out that they don't need to match H and E to learn context-dependent representations. Therefore, for ALBERT, instead of projecting the one-hot vectors of WordPiece embedding into the hidden space of size H, they first project them into embedding of size E, in which E has a lower dimension than H, and then project them into the hidden layer. By doing this, the number of parameters are reduced from (V X H) to (V X E + E X H)[2]. Since the number of hidden units are no longer linked to embedding, ALBERT can be expanded scale on hidden layers without significant increase in parameters.

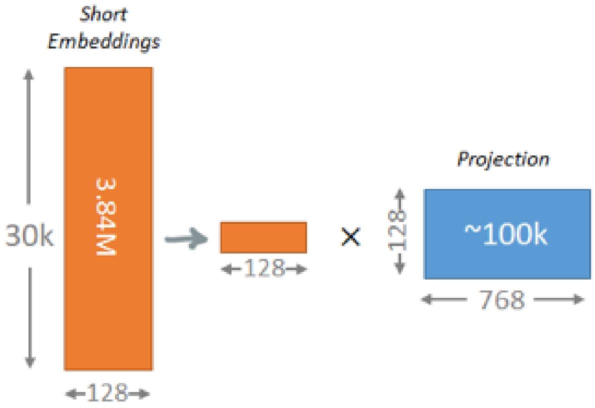
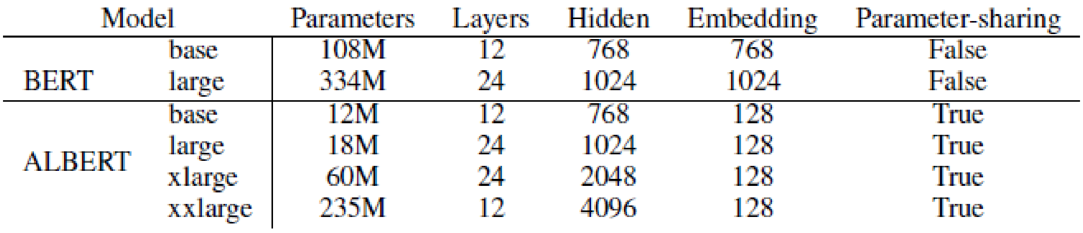


Figure 14: Embedding factorization [13]

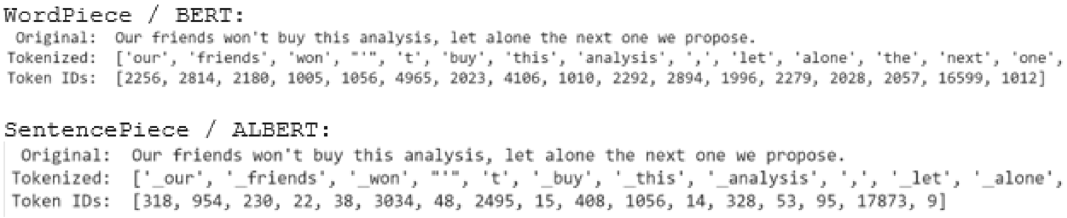
After implementation of two parameter reduction techniques, ALBERT has great reduction on unique parameters. The unique parameters on ALBERT xxlarge model is less than the number on BERT-LARGE. Those techniques also act as regularization to the model to prevent overfitting.



Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Table 2: Configuration of BERT and ALBERT [2]

WordPiece in BERT breaks a word into different tokens. It is hard for the model to recognize which tokens belong to the same word.For better tokenizing subwords, ALBERT uses SentencePiece instead of WordPiece to tokenize word embedding. SentencePiece allows the tokenization to break words in a more connectable way. For tokens from the same word, there is no underline before the word. The SentencePiece helps the model to recognize individual subwords at token level[13].



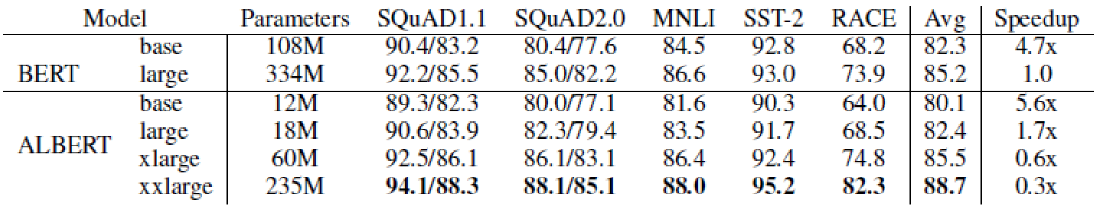
```
WordPiece / BERT:
 Original:  Our friends won't buy this analysis, let alone the next one we propose.
Tokenized:  ['our', 'friends', 'won', "'", 't', 'buy', 'this', 'analysis', ',', 'let', 'alone', 'the', 'next', 'one',
Token IDs:  [2256, 2814, 2180, 1005, 1056, 4965, 2023, 4106, 1010, 2292, 2894, 1996, 2279, 2028, 2057, 16599, 1012]

SentencePiece / ALBERT:
 Original:  Our friends won't buy this analysis, let alone the next one we propose.
Tokenized:  ['_our', '_friends', '_won', "'", 't', '_buy', '_this', '_analysis', ',', '_let', '_alone',
Token IDs:  [318, 954, 230, 22, 38, 3034, 48, 2495, 15, 408, 1056, 14, 328, 53, 95, 17873, 9]
```

Figure 15: Comparison of WordPiece and SentencePiece [13]

Next Sentence Prediction (NSP) is one of the two pretraining tasks on BERT[1]. It is used to train the model to predict whether two sentences are consecutive. However, the ALBERT development team found that the task is unreliable due to easy prediction on sentence topic rather than sentence coherence. The task will therefore focus on topic prediction instead of sentence coherence. Therefore, the team developed a new pretraining task Sentence Order Prediction (SOP) to learn inter-sentence coherence instead of predicting common topics. The SOP task improves the performance of the model on fine tuning downstream tasks with multi-sentences input[2].

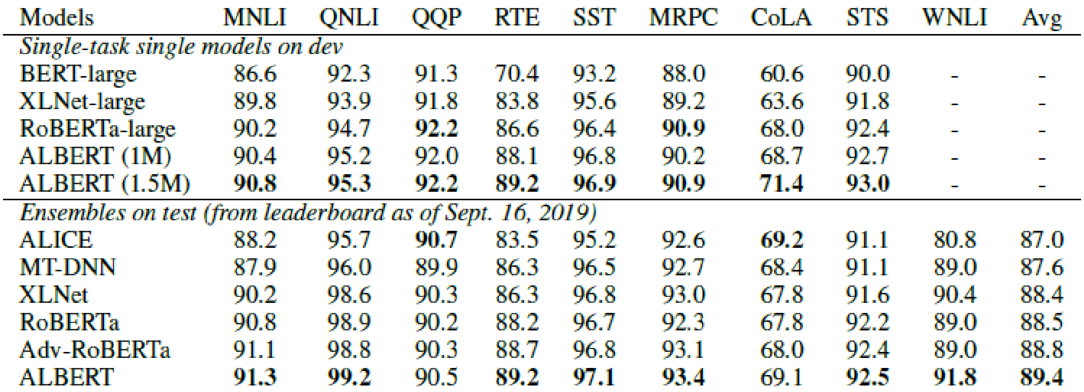
Ngram maximum is another technique used on pretraining tasks of ALBERT. For the Masked Language Model (MLM) task in ALBERT, the model does not only randomly mask a word for prediction. Instead, it randomly masks one to three consecutive words as masked tokens. This improves the predictive ability that the model predicts sequences of words from the text.

Even though there is no improvement in training time on the ALBERT-XXLARGE model compared to BERT-LARGE model, the ALBERT can achieve SOTA performance with fewer unique parameters[2]. The result from the experiment by the development team also shows that the performance of the ALBERT model improved as the training steps increased from 1M to 1.5M. Compared to the BERT model, ALBERT is more capable to achieve better result by training on more steps without facing the problem of GPU/TPU limitation.



Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

Table 3: Dev set results for models pre-trained over BOOKCORPUS and Wikipedia for 125k steps. [2]



Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

Table 4: results of different models on the GLUE benchmark [2]

2.2.5. BioBERT

The biomedical literature continues to grow daily. PubMed alone contains over 29M articles as of Jan 2019. For this reason, there is an increase in the demand of text mining tools that can extract information from these articles. In this paper, researchers aim to use BERT to perform BioNLP related tasks by training it on biomedical corpora.

The original BERT model is pre-trained on English wikipedia and Book corpus. This is a general domain corpus. BERT may not perform well when used in the biomedical domain. The approach that the BioBERT development team took was to train and fine-tune the BERT model on medical corpora, PubMed Abstracts and PMC full text articles, using the weights from original BERT as initialization.

BioBERT fine-tune was done on three tasks: Name entity recognition(NER), Relation extraction(RE), and QA. The NER task contains 9 datasets including datasets related to chemicals, diseases, medical records, protein, and drugs. These datasets include: NCBI disease, i2b2 2010, BC5CDR-disease, BC5CDR-chemical, BC4CHEMD, BC2GM, JNLPBA, LINNAEUS, and Species-800. The RE task contains 3 datasets related protein-chemical, and gene-disease. These datasets include: GAD, EU-ADR, and ChemProt. Last task is QA datasets include: BioASQ 4b, BioASQ 5b, and BioASQ 6b[6].

BioBERT was able to achieve outstanding results in both fine-tuning tasks. In NER, it was able to outperform the SOTA in 2019 by 0.62 in F1 score. For the RE task, the BioBERT was able to outperform SOTA in two out of three datasets. Finally in QA, BioBERT was able to achieve a mean reciprocal rank score 7 higher than SOTA and original BERT model.

2.2.6. BLUEBERT

Over the past decade, there has been an increase in language training representation in the biomedical domain on multiple tasks. Before this paper was released, there has been the general language understanding evaluation benchmark or GLUE which is used for training NLP tasks in the general domain but there hasn’t been any benchmark for the biomedical domain. The BLUEBERT paper introduced the biomedical language understanding evaluation(BLUE) benchmark for NLP researchers in the biomedical field.

BLUE benchmark contains five NLP tasks with ten datasets[7] that have been widely used by the biomedical NLP community as shared tasks.

- ❖ Sentence similarity

Sentence similarity task is to predict similarity scores by estimating whether two sentences deliver similar contents. The datasets belong to this task includes: BIOSSES and MedSTS

- ❖ Named entity recognition

Name entity recognition detects name entities and categorizes entities into different predefined classes. BLUEBERT constructed their dataset using spaCy to split the text into a sequence of tokens. These dataset includes: BC5CDR and Share/CLEF

- ❖ Relation extraction

The aim of relation extraction is to extract semantic relationships between two entities mentioned in a sentence. The datasets belong to this task are: DDI, ChemProt and I2b2 2010

- ❖ Document multilabel classification

Document classification classifies the whole document into various categories. Multiple labels from texts are predicted in the multilabel classification task. The dataset belongs to this task is: HoC

- ❖ Inference task

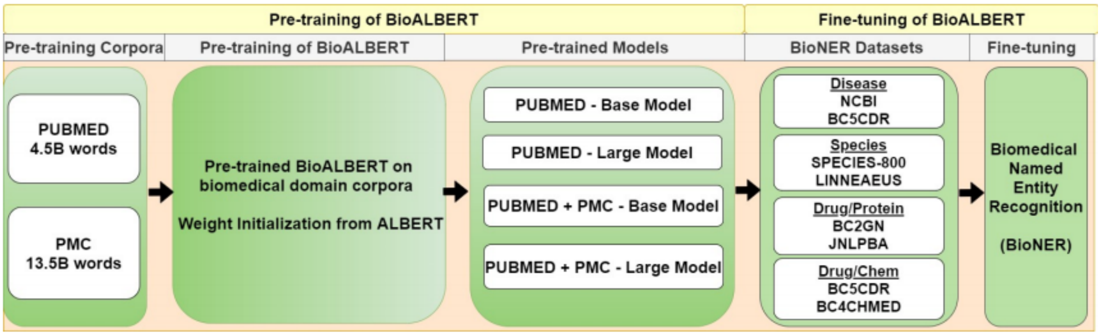
Inference tasks predict whether or not the premise sentence entails the hypothesis. It mainly focuses on causation relationships between sentences. The dataset belongs in this task includes: MedNLI

In order to investigate the effectiveness of BLUE benchmark, experiments were conducted with two baselines: BERT model and ELMo. Both are SOTA models with shown great results in NLP tasks. BERT base and large models were pre-trained on Pubmed abstracts and MIMIC-III clinical notes while ELMo model was pre-trained on Pubmed abstracts only[7].

From the experiment, BERT-base pre-trained on Pubmed abstract and MIMIC III achieved the best results for all five tasks. It is significantly superior to other models including SOTA models in clinical and biomedical domain datasets.

2.2.7. BioALBERT

Biomedical ALBERT(A Lite Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) is an ALBERT model pre-trained on large-scale biomedical corpus, PubMed Abstract and PMC full text articles, for the purpose of performing a total of eight name entity recognition tasks. The table below shows an overview of the pre-training and fine-tuning process of BioALBERT[10].



[Attention based models on Health Informatics Data]

Figure 16: Overview of BioALBERT pre-train and fine-tune[10]

BioALBERT was initialized with weight from the original ALBERT models. They pretrained four ALBERT models on PubMed and PMC. These four models include ALBERT base & large pre-trained with Pubmed abstract and ALBERT base & large pre-trained with PubMed abstract and PMC. After pre-train, these models are fine-tuned on eight NER datasets which includes BC5CDR, BC4CHEMD, NCBI Disease, JNLPBA, BC2GM, LINNAEUS, and lastly Species-800. The results from each dataset are compared with the SOTA model, BioBERT.

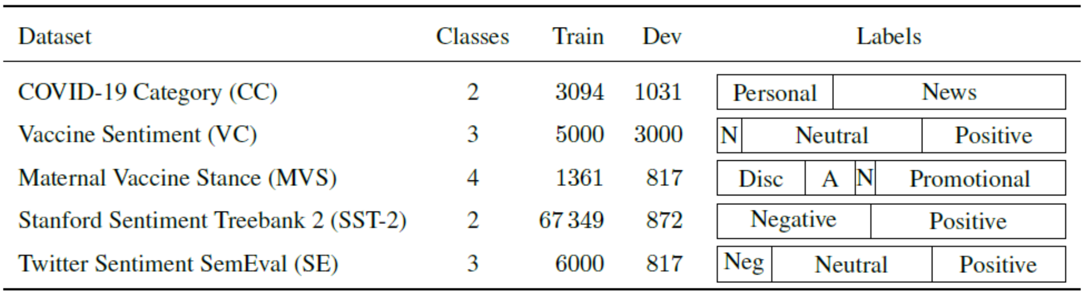
BioALBERT shows superior results than BioBERT. All the results of different BioALBERT pre-trained models are equally as good but BioALBERT large trained on only PubMed abstracts managed to slightly outperform other models.

2.2.8. CTBERT

The CT-BERT model was trained on the corpus of total 160M tweets related to the COVID-19 from the January 12 to April 16 2020[9]. Since Twitter is a public medium that contains a large amount of information and expression from its users, information in specific topics is usually combined with meaningless expression and unstructured content. Therefore, the dataset needed to be cleaned before training. Development team of CTBERT replaced the emojis with their respective ASCII characters, usernames and URL with common tokens. Tweets were then converted to individual documents and sentences by using spaCy.

The model used five independent datasets for downstream classification tasks. Three of them are publicly available related to covid and the two were from their own internal projects. All five fine tuning tasks are sentiment analysis tasks with a number of classes from 2 to 4.

The five dataset used in sentiment analysis are:
COVID-19 Category: subsample of the dataset annotated by Amazon Turk and categorised into personal narrative or news
Vaccine Sentiment (VS): the collection of measles and vaccination related tweets
Maternal Vaccine Stance (MVS): the dataset is annotated by experts into four categories including neutral, discouraging, promotional and ambiguous
Twitter Sentiment SemEval (SE): an unpublished dataset related towards the stance of using maternal vaccines
Stanford Sentiment Treebank 2 (SST-2): a public dataset that consists of binary sentiment labels.



Dataset	Classes	Train	Dev	Labels
COVID-19 Category (CC)	2	3094	1031	Personal News
Vaccine Sentiment (VC)	3	5000	3000	N Neutral Positive
Maternal Vaccine Stance (MVS)	4	1361	817	Disc A N Promotional
Stanford Sentiment Treebank 2 (SST-2)	2	67 349	872	Negative Positive
Twitter Sentiment SemEval (SE)	3	6000	817	Neg Neutral Positive

Table 5: Overview of CT-BERT fine-tuning task[9]

The CT-BERT team achieved an average 17.57% improvement in marginal performance compared to the BERT-LARGE model in the same tasks[9]. It is expected that CT-BERT can perform fairly well on other classification problems from social media platforms. The experimental result from the CT-BERT team also shows that MLM and NSP metrics used by BERT contribute some degree improvement to classification tasks during pre training.

2.3. Summary

In this project, two new models have been created using domain specific datasets. Developing a new BERT based model requires deep understanding of transformer architecture inside each layer and how those models extract contextual information from text. Modified on the architecture of BERT, ALBERT is considered a better option in terms of resource consumption. As mentioned previously, some models originated and developed from the original BERT model. CT-BERT, BLUEBERT and BIOBERT are studied as our reference models. By matching the configuration of those models, the performance of new vaccine BERT and new BioALBERT models become more comparable to existing SOTA models.

3. Research Problem

As previously stated in the introduction section this project takes inspiration from BioALBERT and it aims to pre-train domain specific models specifically on clinical dataset (MIMIC III) which has clinical notes typed by doctors, nurses and other health care workers on 40K patients. These notes are free form notes that contain patient names, diagnosis, names of diseases and the medications given or prescribed to these patients. Although prior research such as BioALBERT [6] and BioBERT[10] have achieved great results however they were trained on very sanitised datasets extracted from research paper extracts. The gap that this research attempts at covering is training the same models as BioBERT and BioALBERT on hand types clinical notes which closely resembles real world scenarios and hence attempting to produce a model that could have a higher accuracy in extracting useful information from real world applications where the input is not sanitised.

3.1. Research Aims & Objectives

The research aims to investigate the effectiveness of current biomedical SOTA models such as BioBERT and BioALBERT on clinical tasks and whether an extended BioALBERT pre-trained with MIMIC III clinical notes data can outperform the current SOTA. This is important as it extends the utility and also tests the robustness of the bioALBERT models.

Hence BioALBERT was trained on downstream tasks namely extraction, named entity recognition, sentence similarity, document classification and inferencing with data from a variety of sources that has been widely used in the Biomedical NLP community as shared tasks. BioBERT was not trained on these tasks as the BLUE benchmark already includes results of experiments on BioBERT trained on clinical tasks.

This project also aimed to train attention based models on vaccine related tweets in addition to the MIMIC-III datasets however due to a bug in tensorflow 2.1 this investigation didn't produce desirable outcomes and will be discussed in the following sections in detail.

3.2. Research Questions

The main research questions this paper is attempting to answer are as follows:

1. How do BioALBERT and BioBERT perform when trained on downstream tasks created from clinical datasets?
2. Can an extended BioALBERT model trained on clinical notes data from real patients (MIMIC-III) outperform the current BioBERT and BioALBERT?

3.3. Research Scope

In order to achieve the aims of the research project, the following scope of work has been conducted.

A) BioALBERT has been pre-trained on the MIMIC-III dataset which is a large database consisting of clinical related data associated with more than 40000 patients producing four models namely which have been further trained on the above mentioned downstream tasks.

1. BioALBERT 2.0 (+ PubMed + MIMIC III 200K) [Base]
2. BioALBERT 2.0 (+ PubMed + PMC + MIMIC III 200K) [Base]
3. BioALBERT 2.1 (+ PubMed + MIMIC III 270K) [Large]
4. BioALBERT 2.1 (+ PubMed + PMC + MIMIC III 270K) [Large]

B) Each of the models have been fine-tuned on downstream tasks listed below:

- Named entity recognition task

- Relation extraction task
- Sentence similarity task
- Inferencing task
- Document classification task

C) BERT has been trained on vaccine tweets and fine-tuned on sentiment analysis downstream tasks and compared against CT-BERT as a benchmark.

4. Methodology

This section presents an end to end machine learning pipeline implemented in this project. Our methodology can be broken down into 7 main sections: (i) data acquisition, sources of each dataset and steps involved in acquiring them, (ii) datasets description, (iii) data pre-processing for pre-training, (iv) pre-training, (v) fine-tuning, (vi) baselines models, and (vii) performance evaluation for fine-tune.

The following diagrams summarised the general approach taken in this project.

Pipeline 1 - BioAlbert + MIMIC-III vs BioBERT vs BlueBERT

Pre-training of Albert models on MIMIC-III datasets and the downstream fine-tuning are shown by the flow chart below.

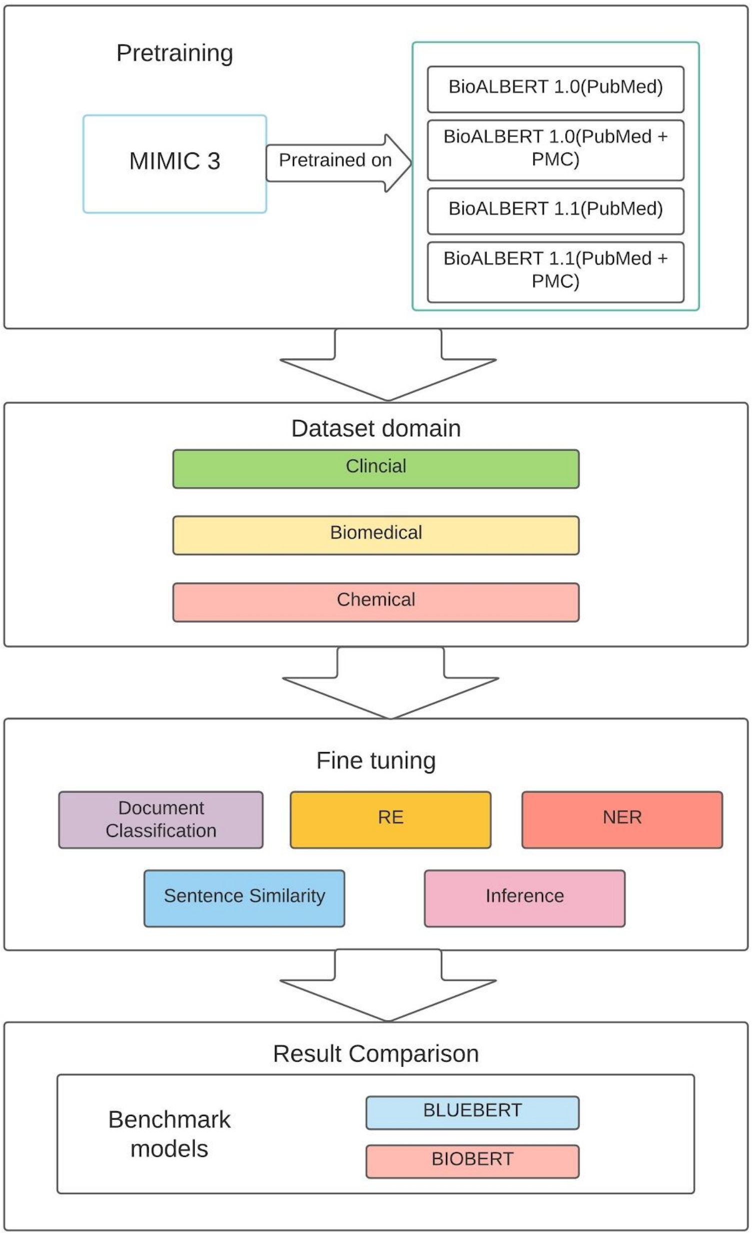


Chart 1: Process flow chart of pretraining ALBERT on MIMIC and fine tuning on downstream tasks

Pipeline 2 - BERT + Vaccine tweets vs CT-BERT

The pre-processing, pre-training and fine-tuning of Bert with vaccine tweets is shown by the following flow chart.

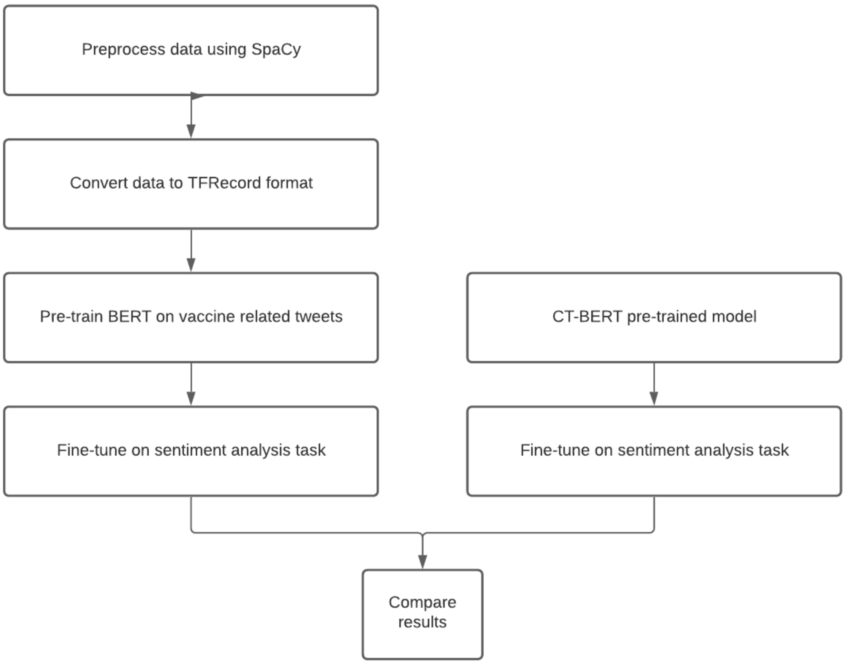

Preprocess data using SpaCy
Convert data to TFRecord format
Pre-train BERT on vaccine related tweets
CT-BERT pre-trained model
Fine-tune on sentiment analysis task
Fine-tune on sentiment analysis task
Compare results

Chart 2: Process flow chart of pretraining and fine tuning BERT on vaccine data

4.1. Data acquisition

The datasets that were used for pre-training and fine-tuning tasks were acquired from a variety of sources some of which were publicly available while others required access from the relevant organisation. The following table is a summary of the dataset and where they were acquired from.

Data Acquisition Table		
Datasets	**Tasks**	**Source**
Chemprot	RE	Available from Bluebert github repo
DDI	RE	Available from Bluebert github Bluebert github repo
i2b2	RE	Applied for access from National Center for Biomedical Computing
HOC	Document classification	Available from Bluebert github Bluebert github repo

BIOSSES	Sentence Similarity	Available from Bluebert github Bluebert github repo
MedNLI	Inference	Applied for access from physionet.org
ShARe/CLEFE	NER	Applied for access from physionet.org
MedSTS	Sentence Similarity	Applied for access from Mayo Clinic
MIMIC-III	Unlabelled clinical dataset for pre-training	Applied for access from physionet.org
Vaccine tweets	Unlabelled vaccine tweets dataset for pre-training	Mr. Usman Naseem
BC2GM	NER	Mr. Usman Naseem
BC4CHEMD	NER	Mr. Usman Naseem
BC5CDR-chem	NER	Mr. Usman Naseem
BC5CDR-disease	NER	Mr. Usman Naseem
JNLPBA	NER	Mr. Usman Naseem
linnaeus	NER	Mr. Usman Naseem
NCBI-disease	NER	Mr. Usman Naseem
s800	NER	Mr. Usman Naseem
euadr	RE	Mr. Usman Naseem
GAD	RE	Mr. Usman Naseem

Table 6: Source of dataset

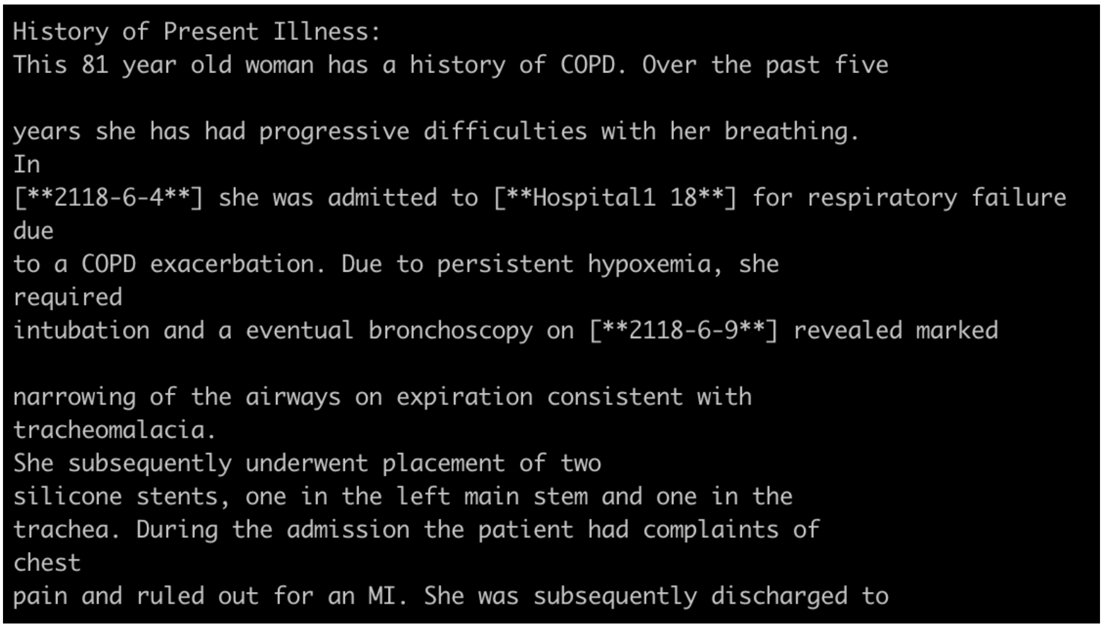
Fine-tuning was acquired from Bluebert's github repository except for i2b2, NLI, ShARe/CLEFE*

4.2. Datasets

4.2.1. Pre-training datasets

4.2.1.1. MIMIC III

MIMIC III(Medical Information Mart for Intensive Care) is a large database consisting of clinical related data associated with more than 40000 patients. It contains detailed information regarding the health records of those patients. The dataset contains many files that have been dumped from a hospital database system. A basic data exploration reveals that clinical notes are located in the NOTEEVENTS.csv file. The following screenshot shows an excerpt from the NOTEEVENTS.csv which contains free form notes typed by doctors, nurses, radiologists and the like.



```
History of Present Illness:
This 81 year old woman has a history of COPD. Over the past five

years she has had progressive difficulties with her breathing.
In
[**2118-6-4**] she was admitted to [**Hospital1 18**] for respiratory failure
due
to a COPD exacerbation. Due to persistent hypoxemia, she
required
intubation and a eventual bronchoscopy on [**2118-6-9**] revealed marked

narrowing of the airways on expiration consistent with
tracheomalacia.
She subsequently underwent placement of two
silicone stents, one in the left main stem and one in the
trachea. During the admission the patient had complaints of
chest
pain and ruled out for an MI. She was subsequently discharged to
```

Figure 17: Clinical Record from MIMIC III[42]

This dataset included 524 million words equivalent to 3.8 GB of data.

4.2.1.2. Vaccine related tweets

This dataset has 64M tweets that are related to vaccines or vaccination. This data has been acquired from twitter and provided by Mr. Usman Naseem.

Following table contains a summary of corpus used for pre-training of the models.

Corpus	Number of words	Domain
MIMIC-III	524M	Clinical
Vaccine related tweets	64M	Biomedical - Vaccine

Table 7: Corpas on pre-training dataset[42]

4.2.2. Fine-tuning datasets

The datasets we used for fine-tuning are extensively used for research in the biomedical NLP community. In this section we'll be going through the datasets used for each of the fine-tuning tasks namely Name Entity Recognition, RE, sentence similarity, document classification, and Inference. There are a total of 8 datasets from BLUEBERT benchmark, 9 datasets from BioBERT and 1 dataset on vaccine related tweets. The table below shows a summary of all the datasets.

Datasets	**Tasks**	**Domain**	**Train**	**Dev**	**Test**
BC5CDR-Di	NER	Clinical	109853	121971	129472
BC5CDR-Ch	NER	Chemical	109853	117391	124676
NCBI Disease	NER	Clinical	135615	23959	24488
JNLPBA	NER	Biomedical	443653	117213	114709
BC2GM	NER	Biomedical	333920	70937	118189
LINNAEUS	NER	Biomedical	267500	87991	134622
Species-800	NER	Biomedical	147269	22217	42287
GAD	RE	Biomedical	3277	1025	820
euadr	RE	Biomedical	227	71	57
ShARe/CLEFE*	NER	Clinical	4628	1075	5195
DDI*	RE	Biomedical	2937	1004	979
ChemProt*	RE	Biomedical	4154	2416	3458
i2b2 2010*	RE	Clinical	3110	11	6293

HoC*	Document Classification	Biomedical	1108	157	315
MedNLI*	Inference	Biomedical	11232	1395	1422
MedSTS*	Sentence Similarity	Clinical	675	75	318
BIOSSES*	Sentence Similarity	Biomedical	64	16	20
Vaccine related tweets	Sentiment analysis	Vaccine	11135	3664	3665

*Table 8: Datasets from BioBERT and dataset with * from BLUEBERT[6][7]*

4.2.2.1. NER (Name Entity Recognition)

BC5CDR(Disease and Chemical) - This dataset contains 1500 PubMed titles and abstracts that have been selected from Pfizer datasets that was used in the BioCreative V chemical and disease task, which is composed of 4409 annotated chemicals, 5818 diseases and 3116 chemical-disease interactions. It's a dataset that describes the relationship between Chemical and Disease concepts.

JNLPBA - Its a dataset that consists of 2000 abstracts from the GEnia version 3 MEDLINE abstracts and requires the identification of bio entities of interest in microbiology for the Name Entity Recognition task.

LINNAEUS - It is an annotated collection that consists of 153 Pubmed articles that has a total collection of 4077 species.

NCBI-Disease - It is a fully annotated corpus that has 6892 disease mentions from the 793 Pubmed abstracts and 790 unique disease concepts that identifies the disease mentioned in the Pubmed abstracts.

Species 800 - It comprises 800 Pubmed abstracts which contains the mentions of organisms which were identified and mapped with a focus on their species.

BC2GM - It contains gene mentions and protein-protein interactions with 20,000 sentences which consist of 4,300 gene annotations.

Share/CLEFE- It contains 300 de-identified clinical notes from the MIMIC III dataset. The disorder mentions were annotated by two professional coders who were trained for this task.

4.2.2.2. RE (Relation Extraction)

I2B2 2010 - Its a shared task from the challenge in NLP for clinical data and contains 170 documents for training and 256 documents for testing. This corpus is a collection of datasets from three different hospitals, annotated by medical practitioners for eight types of relationships.

Chemprot - It consists of chemical protein interaction which were annotated by experts from the 1820 Pubmed abstracts which was used in the BioCreative Vi challenge.

DDI - It's an annotated dataset that consists of drug-drug interactions and other pharmacological substances annotated by experts. It has 792 texts from the DrugBank dataset and 233 Medline abstracts.

GAD - It is a database of complex diseases and disorders data from the Genetic Association Database. Records in this dataset are annotated with links to other reference databases such as PubMed and CDC.

Euadr - It is a dataset that consists of annotation for drugs, disorders, genes and their inter-relationships. Each of the drug–disorder, drug–target, and target–disorder relation pairs have a set of 100 abstract by three expert annotators.

4.2.2.3. Sentence similarity

BIOSSES - sentence pairs corpus that contains 100 pairs of sentences from the Biomedical Summarization Track Training Dataset which are manually annotated by five experts. Sentence pairs are rated by similarity scores from 0 to 4.

MedSTS - sentence pairs corpus from Mayo Clinic annotated by two medical experts. The semantic similarity scores are ranked from 0 to 5 which indicate similarity from low to high.

4.2.2.4. Document Classification

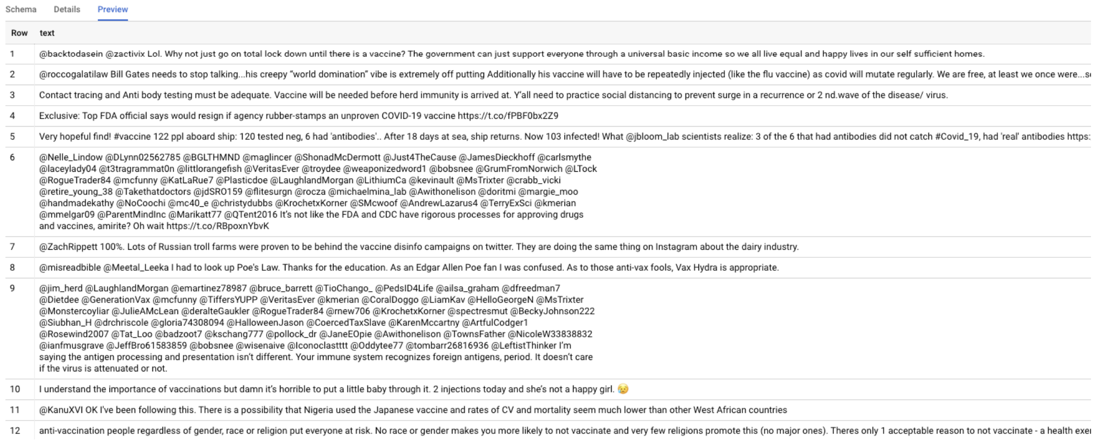
HOC - The Hallmarks of Cancer (HOC) Corpus contains 1580 PubMed publication abstracts manually created by experts according to the Hallmarks of Cancer taxonomy. This dataset retrieves cancer-related references from PubMed. A sentence in this dataset may belong to zero or more classes.

4.2.2.5. Inference

MedNLI - This dataset contains sentence pairs from MIMIC-III patient's medical history which was specially annotated by doctors. The sentence pairs contain one premise sentence and hypothesis sentence. It is used to predict whether the premise shows entailment, contradiction or neutral.

4.2.2.6. Sentiment analysis

Vaccine related tweets - This dataset is vaccine related tweets from twitter. It contains a large amount of unstructured expressions or slang messages. Special tags such as emojis, URLs and username made up a large proportion of the dataset.



Row	text
1	@backtodasein @zactivix Lol. Why not just go on total lock down until there is a vaccine? The government can just support everyone through a universal basic income so we all live equal and happy lives in our self sufficient homes.
2	@roccogalatilaw Bill Gates needs to stop talking...his creepy “world domination” vibe is extremely off putting Additionally his vaccine will have to be repeatedly injected (like the flu vaccine) as covid will mutate regularly. We are free, at least we once were...s
3	Contact tracing and Anti body testing must be adequate. Vaccine will be needed before herd immunity is arrived at. Y’all need to practice social distancing to prevent surge in a recurrence or 2 nd.wave of the disease/ virus.
4	Exclusive: Top FDA official says would resign if agency rubber-stamps an unproven COVID-19 vaccine https://t.co/fPBF0bx2Z9
5	Very hopeful find! #vaccine 122 ppl aboard ship: 120 tested neg, 6 had 'antibodies'.. After 18 days at sea, ship returns. Now 103 infected! What @jbloom_lab scientists realize: 3 of the 6 that had antibodies did not catch #Covid_19, had 'real' antibodies https:
6	@Nelle_Lindow @DLynn02562785 @BGLTHMND @maglincer @ShonadMcDermott @Just4TheCause @JamesDieckhoff @carlsmythe @laceylady04 @t3tragrammat0n @littlorangefish @VeritasEver @troydee @weaponizedword1 @bobsnee @GrumFromNorwich @LTock @RogueTrader84 @mcfunny @KatLaRue7 @Plasticdoe @LaughlandMorgan @LithiumCa @kevinault @MsTrixter @crabb_vicki @retire_young_38 @Takethatdoctors @jdSRO159 @flitesurgn @rocza @michaelmina_lab @Awithonelison @doritmi @margie_moo @handmadekathy @NoCoochi @mc40_e @christydubbs @KrochetxKorner @SMcwoof @AndrewLazarus4 @TerryExSci @kmerian @mmelgar09 @ParentMindInc @Marikatt77 @QTent2016 It’s not like the FDA and CDC have rigorous processes for approving drugs and vaccines, amirite? Oh wait https://t.co/RBpoxnYbvK
7	@ZachRippett 100%. Lots of Russian troll farms were proven to be behind the vaccine disinfo campaigns on twitter. They are doing the same thing on Instagram about the dairy industry.
8	@misreadbible @Meetal_Leeka I had to look up Poe's Law. Thanks for the education. As an Edgar Allen Poe fan I was confused. As to those anti-vax fools, Vax Hydra is appropriate.
9	@jim_herd @LaughlandMorgan @emartinez78987 @bruce_barrett @TioChango_ @PedsID4Life @ailsa_graham @dfreedman7 @Dietdee @GenerationVax @mcfunny @TiffersYUPP @VeritasEver @kmerian @CoralDoggo @LiamKav @HelloGeorgeN @MsTrixter @Monstercoyliar @JulieAMcLean @deralteGaukler @RogueTrader84 @rnew706 @KrochetxKorner @spectresmut @BeckyJohnson222 @Siubhan_H @drchriscole @gloria74308094 @HalloweenJason @CoercedTaxSlave @KarenMccartny @ArtfulCodger1 @Rosewind2007 @Tat_Loo @badzoot7 @kschang777 @pollock_dr @JaneEOpie @Awithonelison @TownsFather @NicoleW33838832 @ianfmusgrave @JeffBro61583859 @bobsnee @wisenaive @Iconoclastttt @Oddytee77 @tombarr26816936 @LeftistThinker I’m saying the antigen processing and presentation isn’t different. Your immune system recognizes foreign antigens, period. It doesn’t care if the virus is attenuated or not.
10	I understand the importance of vaccinations but damn it’s horrible to put a little baby through it. 2 injections today and she’s not a happy girl. 😥
11	@KanuXVI OK I've been following this. There is a possibility that Nigeria used the Japanese vaccine and rates of CV and mortality seem much lower than other West African countries
12	anti-vaccination people regardless of gender, race or religion put everyone at risk. No race or gender makes you more likely to not vaccinate and very few religions promote this (no major ones). Theres only 1 acceptable reason to not vaccinate - a health exe

Figure 18: preview of vaccine related tweets dataset

4.3. Data pre-processing

In this step data sources are cleansed, normalised then converted to TFRecord format so that they can be ingested by the models in the pre-training step. In this project two separate machine learning pipelines have been constructed to perform two separate experiments with very different objectives, each of these requiring very different pre-processing steps which has been described in the following sections.

Data cleaning/normalisation → Convert to TFRecord format→ Pretrain→ Fine-tune

4.3.1. MIMIC III

In order to prepare the dataset for pre-training this data has to be converted to TFRecord format however this process is very resource intensive and requires quite a lot of RAM. This is because the create_pretraining_data.py script that was provided in the BLUEBERT github repository only runs on a single core and it takes roughly 6 mins to process 1MB of data.

In order to overcome this the dataset was split into 1MB chunks and four batches of 956 files were created and distributed across four 16-core compute optimized vms on google cloud. This reduced the theoretical processing time from 14 days to 1hr. The figure below shows an overview of the preprocessing flow used to distribute the workload across a cluster of four google compute engines.

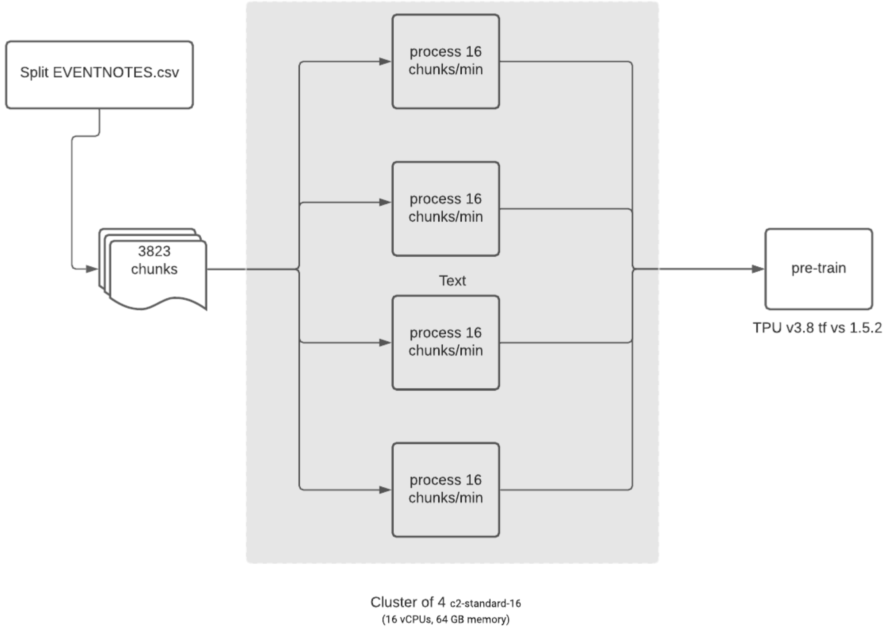


Chart 3: Procedure of pre training BioALBERT on MIMIC III

Size in MB	**Total time to process @ 6min/MB**	**Total RAM required**	**Disk space required**
3380	14.08333333 days	3545.94 GB or ~3.6 TB	~3.6 TB

Table 9: the original pre-processing time of the MIMIC dataset on a single core machine.

4.3.2. Vaccine tweets

The following data processing steps were applied to the vaccine tweets dataset, this was inline with the CT-BERT paper[9] as it was used as a baseline for this experiment.

- **Data deduplication**: Since the data volume was very large google's bigquery was leveraged to help in deduplicating the data. This reduced the data from 64M rows to 19M rows.
- **Converting emojis to ascii**: We used the SpaCy[11] library to convert the emojis into its ascii characters.
- **Filter twitter user names** and replace them with the string <twitteruser>
- **Filter urls** and replace them with the string <twitterurl>
- **If multiple usernames** and urls appear in a single twitter message then replace them with the a single string "twitteruser" and "twitterurl" respectively
- **Remove unicode symbols**

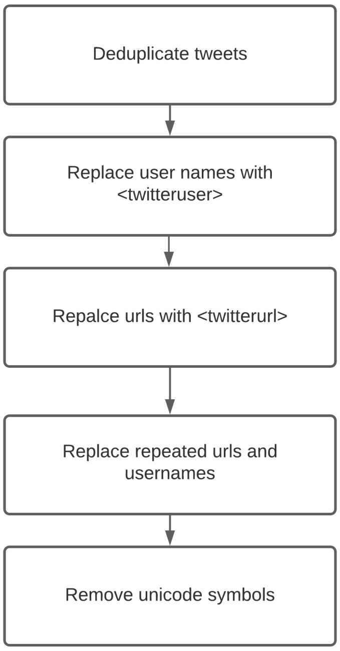

Deduplicate tweets
Replace user names with <twitteruser>
Repalce urls with <twitterurl>
Replace repeated urls and usernames
Remove unicode symbols

Chart 4: Step of preprocessing on tweets data

4.4. Pre-training

The pretraining step involves feeding BERT and ALBERT with unlabeled data such as plain text from, pubmed, pmc, MIMIC-III or vaccine tweets. The pre-training step takes the given data in TFRecord format and trains the BERT/ALBERT model on two fake tasks namely[1]

1. **Masked Language Model (MLM)**
2. **Next Sentence Prediction (NSP)**

This helps BERT/ALBERT to develop sophisticated understanding of the given corpus. Once this pre-training is done, the final layer specific to these tasks is discarded and replaced with a layer specific to the task at hand for fine-tuning.

4.4.1. BioALBERT 2.0 & 2.1

The pre-training step is by far the most compute intensive and costly part of the process, hence for both pre-training jobs google cloud TPU v3-8 (preemptible) were chosen as it is the most performant and cost effective resource. The main consideration is the amount of total memory as pre-training requires quite a lot of memory for larger batch sizes. We chose a batch size of 1024 for the base model and a batch size of 256 for the large model to remain consistent with the bioalbert model.

TPU Type	TPU cores	Total Memory	On demand price (USD)	preemptible price (USD)
v2-8	8	64 GB	$4.5/hr	$1.35/hr
v3-8	8	128 GB	$8/hr	$2.4/hr

Table 10: TPU configuration on BioALBERT pretraining

Model Name	Data	Run - Time (hrs)	Steps
BioALBERT 2.0 Base	PubMED + MIMIC	101	200K
BioALBERT 2.0 Base	PubMED + PMC + MIMIC	107	200K
BioALBERT 2.1 Large	PubMED + MIMIC	111	270K
BioALBERT 2.1 Large	PubMED + PMC + MIMIC	112	270K

Table 11: the run statistics of the pre-trained models

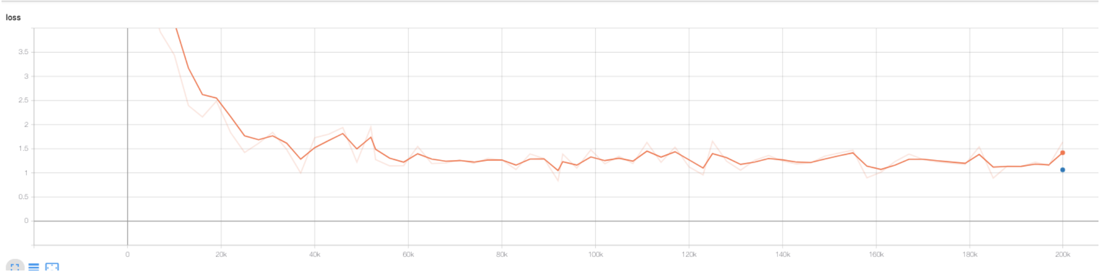


Figure 19: The logloss for BioAlbert 2.0 (PubMED + MIMIC)

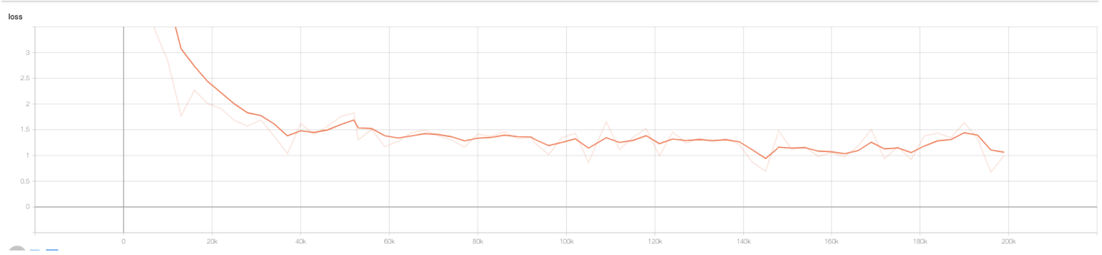


Figure 20: The logloss for BioAlbert 2.0 (PubMed + PMC + MIMIC)

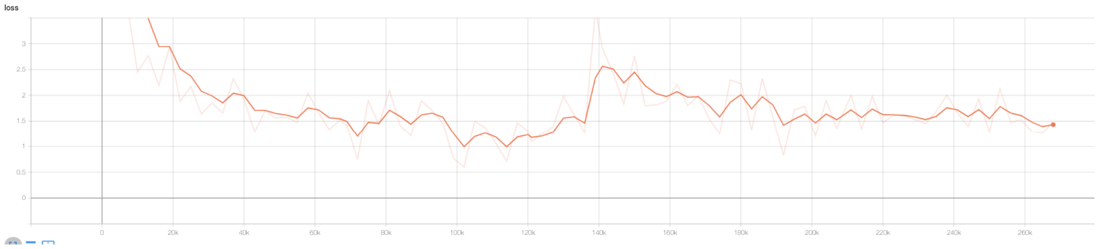


Figure 21: The loglosss for BioAlbert 2.1 (PubMed + MIMIC)

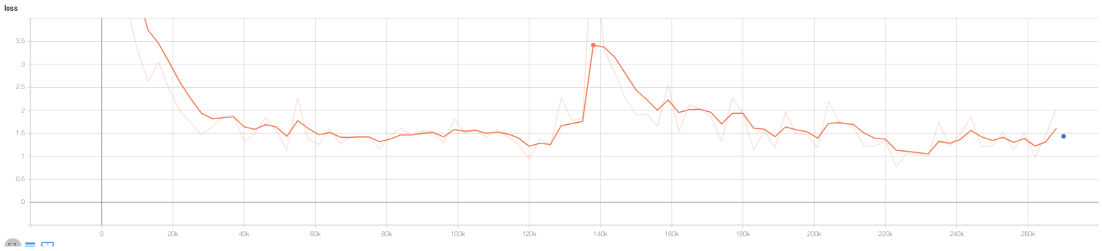


Figure 22: The logloss for BioAlbert 2.1 (PubMed + PMC + MIMIC)

Name	BioAlbert 2.0	BioAlbert 2.1
Architecture	Albert Base	Albert Large
Dropout	0	0
Activation function	GeLU	GeLU
Number of attention heads	12	16
Number of Layers	12	24
Hidden size	768	1024
Embedding size	128	128
Vocab size	30,000	30,000
Optimizer	LAMB	LAMB
Train batch size	1024	256
Eval batch size	16	16
Max sequence length	512	512
Max predictions/seq length	20	20
Learning rate	0.00062	0.00062
Training step (PubMed + MIMIC)	200K	270K

Training step (PubMed + PMC + MIMIC)	200K	270K
Warmup steps	3125	3125

Table 12-a: the experimental setup for pre-training BioALBERT on MIMIC III

4.4.2. Vaccine-BERT

In order to pre-train the BERT model with vaccine related tweets the following experimental settings were used. To pre-train the model preemptible google cloud TPU V3-8 were employed, google cloud TPUs get pre-empted after 24 hours of usage if preemptible option is selected, this requires manual resumption of the model training. We were unable to resume the pre-training after the TPUs were preempted due to a bug in tensorflow 2.1 which wouldn’t allow resumption of pre-training.

Name	**Vaccine BERT**
Architecture	Bert large uncased wwm
Dropout	0.1
Activation function	GeLU
Number of attention heads	16
Number of Layers	24
Hidden size	1024
Embedding size	128
Vocab size	30522
Train batch size	1024
Eval batch size	1024
Max sequence length	96
Max predictions/seq length	20
Learning rate	2e-5
Training step (PubMed + MIMIC)	200K

Table 12-b: the experimental setup for pre-training step on Vaccine BERT

4.5. Fine-tuning

Fine-tuning refers to the final training step, this is because the amount of training required to adapt BERT/ALBERT to a specific task is very small compared to the model pre-training step. The below illustration shows how the same model can be utilized to perform various tasks.

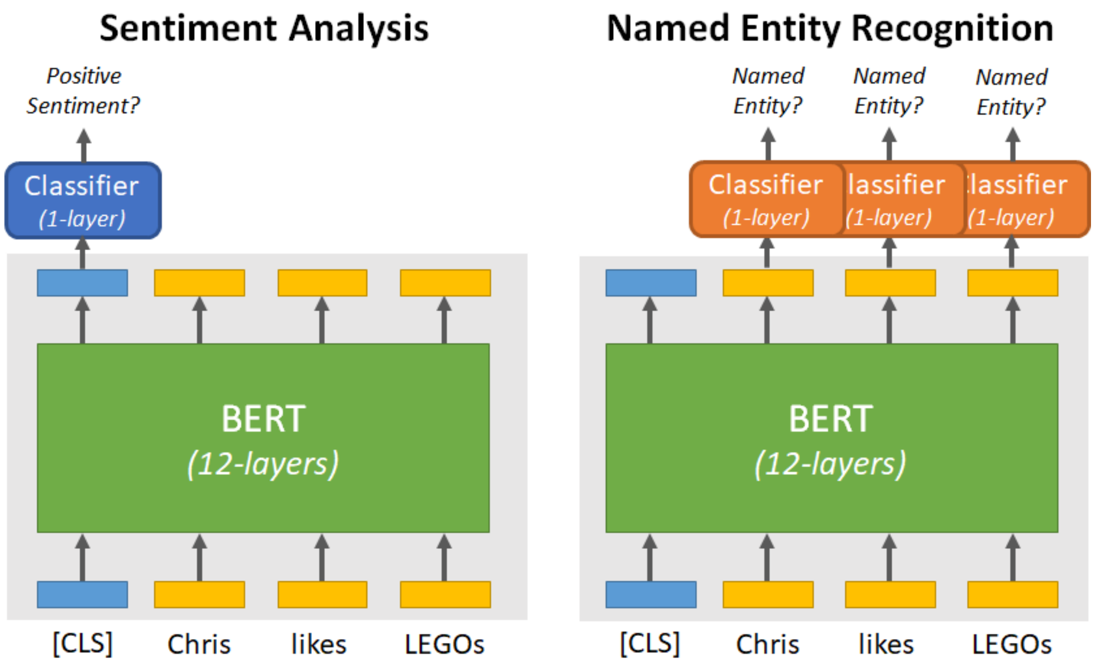


Figure 23: Fine Tuning tasks on BERT (McCormick, C., 2020)

The next sections contain details of the fine-tuning tasks performed in this project.

4.5.1. Fine-tuning of BioALBERT

In this section, the steps involved in fine-tuning BioALBERT models are presented. Our aim for fine-tuning is to extend the fine-tuning task of the original BioALBERT 1.0 & 1.1 models which were only NER, and RE and compare with our new BioALBERT 2.0 & 2.1 models. All BioALBERT were fine-tuned on the BLUE benchmark. In order to further investigate the effectiveness of our new models, BioALBERT 2.0 & 2.1 was fine-tuned on BioBERT datasets. Datasets from both sources have been widely used in the NLP biomedical field. The overview of fine-tuning process.

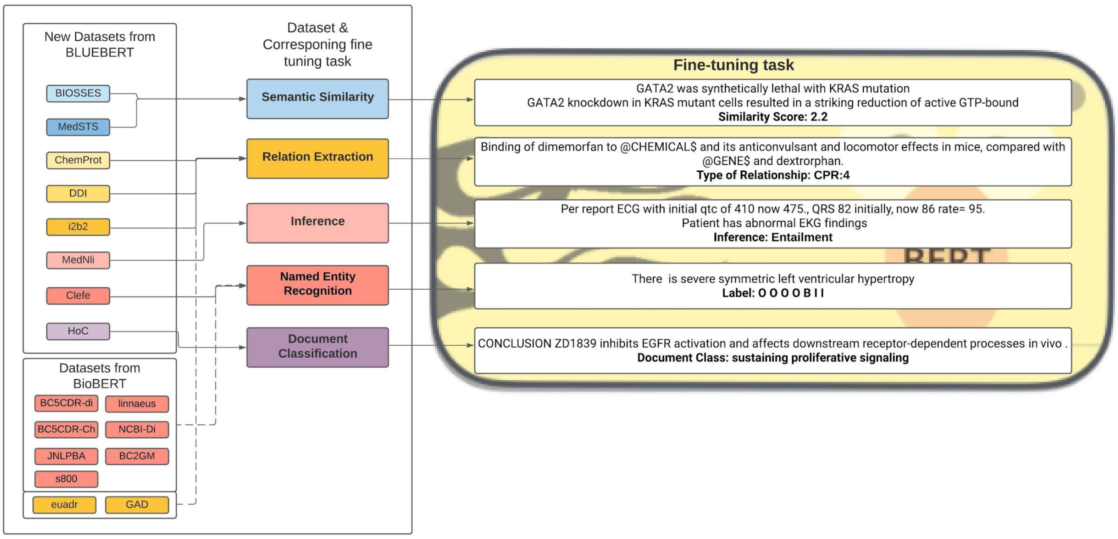

New Datasets from BLUEBERT
BIOSSES
MedSTS
ChemProt
DDI
i2b2
MedNli
Clefe
HoC
Datasets from BioBERT
BC5CDR-di
linnaeus
BC5CDR-Ch
NCBI-Di
JNLPBA
BC2GM
s800
euadr
GAD
Dataset & Corresponing fine tuning task
Semantic Similarity
Relation Extraction
Inference
Named Entity Recognition
Document Classification
Fine-tuning task
GATA2 was synthetically lethal with KRAS mutation
GATA2 knockdown in KRAS mutant cells resulted in a striking reduction of active GTP-bound
Similarity Score: 2.2
Binding of dimemorfan to @CHEMICAL$ and its anticonvulsant and locomotor effects in mice, compared with @GENE$ and dextrorphan.
Type of Relationship: CPR:4
Per report ECG with initial qtc of 410 now 475., QRS 82 initially, now 86 rate= 95.
Patient has abnormal EKG findings
Inference: Entailment
There is severe symmetric left ventricular hypertropy
Label: O O O O B I I
CONCLUSION ZD1839 inhibits EGFR activation and affects downstream receptor-dependent processes in vivo .
Document Class: sustaining proliferative signaling

Figure 24: Overview of BioALBERT & BioALBERT-M fine-tuning process

To set up fine-tuning, the weights of the pre-trained models were used. We set 3e-5 learning rate, batch size to 32, warm up step is 320 and fine-tuned for 10000 train steps. All of the hyperparameters used are the same except for Name Entity Recognition datasets which use 512 max sequence length. For evaluation, the models were saved at every 500 steps in order to investigate the performance at different training steps to determine the best model for making predictions on the test set. Table below shows the summary of the fine-tune setting.

Parameters	Name Entity Recognition only	Other tasks
Train Batch Size	32	32
Optimizer	adamw	adamw
Max seq length	512	128
Learning rate	3e-5	3e-5
Train steps	10k	10k
Warmup steps	320	320
save checkpoints steps	500	500

Table 13: Summary of the fine-tune setting

All of our fine-tuning tasks were run on colab TPU as it is free of charge. However, due to the TPU usage limit, multiple google accounts have to be created in order to complete all of the tasks.

4.5.1.1. Name Entity Recognition

NER(Name Entity Recognition) is NLP tasks that detects named entities and categorizes entities into different predefined classes such as person names, locations, and organizations from a text. NER can enable researchers to find relevant articles or journals by summarizing main ideas and keywords. It is one of the most essential NLP tasks. Many popular NLP models used this task for performance evaluation.

In order to start fine-tuning on NER, firstly pre-process has to be done on the datasets. Since the main aim for fine-tune is to compare the predicted results to baseline models, the same preprocessing technique as the BLUEBERT benchmark was used. BLUEBERT code was used to split the raw data files to tokens and label the word in each sentence into three categories: “O”, ”B”, and “I”. Table below shows the structure of the pre-processed NER dataset.

She	has	stable	lower	extremity	edema
O	O	O	B	I	I

Table 14: Structure of pre-processed Name Entity Recognition dataset

Corpus	B	I	O	Total
BC5CDR-Di	3907	2721	103225	109853
BC5CDR-Ch	4816	1780	103257	109853
NCBI Disease	5130	6111	124374	135615
JNLPBA	32178	64790	346685	443653
BC2GM	14282	20734	298904	333920
LINNAEUS	2036	1087	264377	267500
Species-800	2557	3310	141402	147269
Share/Clefe	4628	4187	90882	99697

Table 15: Distribution of Train Name Entity Recognition dataset

The figure above shows the label count for all NER dataset. These datasets are imbalanced. The count of O class is much higher than B and I.

4.5.1.2. RE

RE(Relationship Extraction) is the task for extracting semantic relationships from a text document. A sentence or paragraph usually contains multiple entities such as person, location, organisation. RE extracts the relationship within those informations. In our case, RE is used to discover relationships in the biomedical text corpus. It detects relationship pairs between various diseases, drugs, symptoms, and viruses.

To fine-tune, document files were converted into sentences with many entity pairs, then entity pairs were masked using masked tokens. There may be multiple entities in a sentence but for each training sample, only one pair of entities is masked and leave others unchanged. Each sentence is labeled with semantically related classes or a false class. For example; Chemprot contains 5 CPR classes and a false class. The table below shows chemprot samples after preprocessing.

16554356.T3.T10	@CHEMICAL$ was chemically bound via linkers to @GENE$-loaded HSA-NP.	FALSE
16554356.T3.T11	Apolipoprotein E was chemically bound via linkers to @CHEMICAL$-loaded @GENE$-NP.	FALSE
16554356.T3.T13	Discovery and optimization of @CHEMICAL$ as inhibitors of methionine aminopeptidase-2: a structural basis for the reduction of @GENE$ binding.	CPR:4
16554356.T3.T14	Discovery and optimization of @CHEMICAL$ as inhibitors of @GENE$: a structural basis for the reduction of albumin binding.	FALSE

Table 16: Structure of preprocessed chemprot dataset

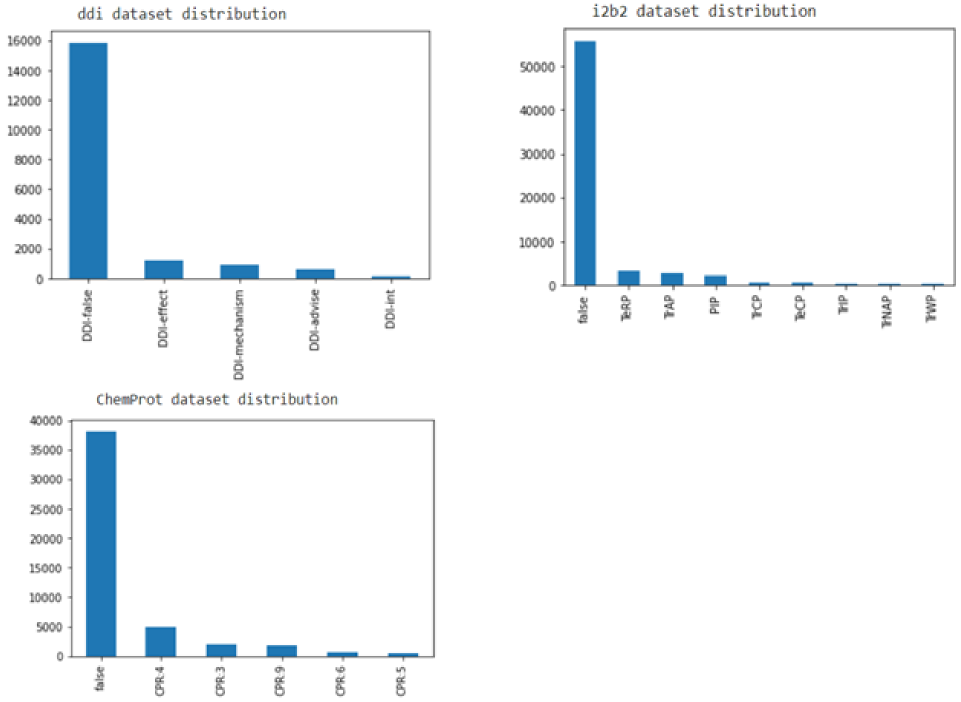


Figure 25: Distribution of labels in RE datasets from BLUEBERT

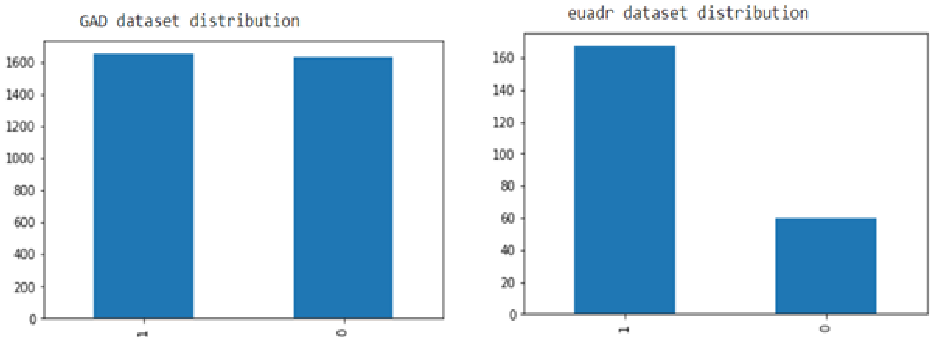


Figure 26: Distribution of labels in RE datasets from BioBERT

From the figure above, our datasets are highly imbalanced. Majority of the sentences belong in the false class. Before evaluating the prediction results, rows with false classes are removed in the test data. Then compare the test labels that contain no false class with predicted labels by matching the index. This is the exact step, used in BLUEBERT RE evaluation procedure.

BioBERT RE datasets have different labels than BLUEBERT. BioBERT classifies the relationship into two types; 1 for related and 0 for not related. These datasets have significant numbers of samples in each class.

4.5.1.3. Sentence Similarity

Sentence similarity task is a task for estimating whether two given sentences or paragraphs deliver similar contents. Sentence similarity measures not only lexical meaning but also the semantic meaning. Given two sentences, the higher is the similarity score, the closer are the sentences in terms of semantic meaning. It focuses on contextual information as a whole rather than the degree of similarity on single words. Therefore, this task allows the user to compare the concept, ideas, or content behind two sentences.

The sentence similarity task is fine-tuned using BIOSSES and MedSTS datasets. These two datasets have similar structure. The only difference is that similarity scores of BIOSSES are graded from 0 to 4 instead of 0 to 5 like MedSTS. O indicates no similarity and the highest score means the two sentences are the same or have the same meaning. The following table shows the structure of STS datasets.

Insulin NPH Human [NOVOLIN N] 100 unit/mL suspension subcutaneous as directed by prescriber.	Insulin NPH Human [NOVOLIN N] 100 unit/mL suspension 63-76 units subcutaneous as directed by prescriber.	3.5
Peripheral IV site, established in the right forearm, using an 18 gauge catheter, in one attempt.	Peripheral IV site, present prior to arrival, established in the right hand, using a 20 gauge catheter.	3.45

Table 17 : Structure of STS datasets

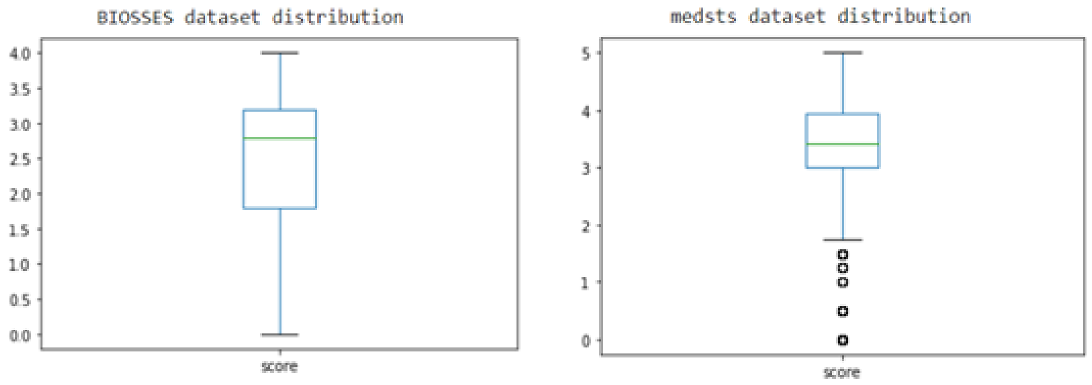


Figure 27: BIOSSES and MedSTS score distribution

Figure above shows the score distribution of BIOSSES and MedSTS datasets. Most scores are in the upper half of the graph, which means that in both datasets most of the sentences are similar to each other.

4.5.1.4. Inference

Given texts of premise and hypothesis, NLP inference can identify whether the premise entails the hypothesis. Unlike sentence similarity tasks, inference concentrates on the causation relationship, not a bidirectional relationship. This task will address the problem in three ways, namely positive, negative and neutral. It can be used to extract causations of phenomenon and correctness of theories. In the biomedical field, one of the uses of this task is to diagnose what causes the symptom of a disease.

Inference task is similar to sentence similarity task in comparing sentences except it is more similar to classification task that classified sentence pairs into three types entailment, contradiction, and neutral. For our fine-tuning, the MedNLI dataset was used, which is a collection of premise and hypothesis sentences pairs extracted from clinical notes MIMIC-III. The following table shows samples from the MedNLI dataset.

In the ED, initial VS revealed T 98.9, HR 73, BP 121/90, RR 15, O2 sat 98% on RA.	The patient is hemodynamically stable	Entailment
In the ED, initial VS revealed T 98.9, HR 73, BP 121/90, RR 15, O2 sat 98% on RA.	The patient is hemodynamically unstable.	Contradiction
In the ED, initial VS revealed T 98.9, HR 73, BP 121/90, RR 15, O2 sat 98% on RA.	The patient is in pain.	Neutral

Table 18: Structure of MedNLI dataset

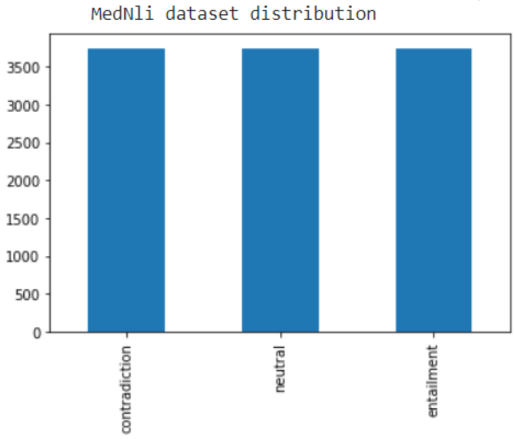


Figure 28: MedNLI label distribution

Figure above shows an example of the distribution of labels in the MedNLI. The labels are equally balanced.

4.5.1.5. Document classification

Document classification is a task similar to Name Entity Recognition except it classifies the whole document into various categories. Text documents are classified into predetermined types. By classifying documents, users can manage and search text in specific domains in a more efficient manner. This task can help users easily build a database for medical use.

The dataset used in this task is HoC(the Hallmarks of Cancers corpus). Each sample in HoC may belong to many classes. Therefore, multilabel classification has to be performed to predict multiple labels from the texts. The table below shows samples of HoC dataset.

11872299_s10	A 15-Gy dose reduced constitutive expression of cyclin D1 in the radiosensitive OCa-I tumors , but had no influence on expression of cyclin D1 in the radioresistant SCC-VII tumors.	sustaining proliferative signaling
11872299_s11	In contrast , 15 Gy increased the expression of p27 in radiosensitive tumors and reduced it in radioresistant tumors.	evading growth suppressors
11872299_s12	Radiation induced no significant apoptosis or change in the percentage of PCNA-positive (proliferating) cells in SCC-VII tumors with high cyclin D1 levels , but it induced significant apoptosis and a decrease in the percentage of proliferating cells in OCa-I tumors with low cyclin D1 expression.	resisting cell death,sustaining proliferative signaling

Table 19: Structure of HoC dataset

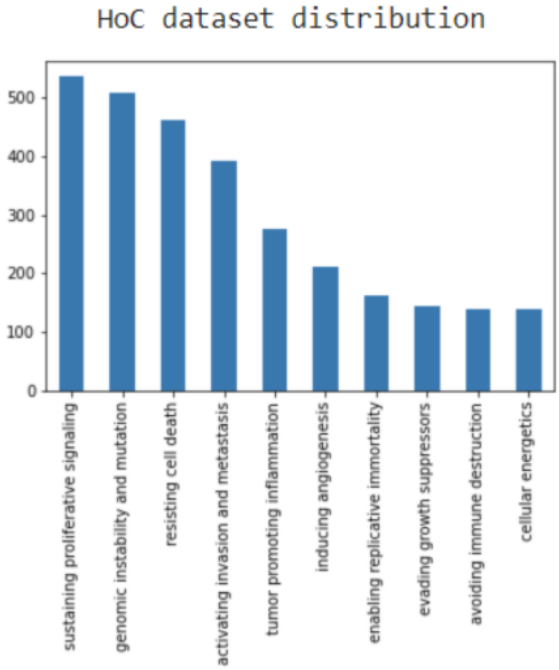


Figure 29: HoC 10 highest count labels distribution

Figure above shows an example of label distribution of HoC dataset. The labels are multi-label. However, the labels with the top 10 highest counts are single labels.

4.5.2. Fine-tuning of Vaccine-BERT

The vaccine tweets were fine-tuned using the code provided on the CT-BERT github repository by the authors of the paper. The following experimental settings were used to fine-tuned both vaccine BERT and CT-BERT with sentiment analysis data.

Parameters	Setting
Train Batch Size	32
Eval batch size	8
Learning rate	2e-5
Train steps	5K

Table 20: Summary of Vaccine-BERT fine-tune setting

4.5.2.1. Sentiment Analysis

Sentiment analysis is a classification task that identifies the polarity of the sentiment from given text and is usually used on detecting the emotion of expressions and comments on social media. Information on Tweeter is often unstructured and Tweets may contain unrelated words to the topic. The data may include username, emojis, URL links or other meaningless words. Therefore, those special words or tags need to be first replaced by comment tokens such as #Username and #emoji using spaCy. To train a model on detecting the sentiment polarity, three values -1, 0 and 1 are given to each Tweet to represent negative, neutral and positive sentiment as labels respectively. Fine tuning Vaccine-BERT on this task allows it to extract sentiment of given tweets and to identify whether those tweet information is positive to vaccine development.

4.5.3. Baseline Models

In this section, the baseline models for BioALBERT and Vaccine-BERT fine-tuning are discussed.

4.5.3.1. Baseline Models for BioALBERT fine-tuning

In order to test the effectiveness of our models, the performance of BioALBERT has to be compared to baseline models. The baseline models chosen to compare our fine-tuning results with are the current SOTA models, BLUEBERT and BioBERT. In this section, the baseline models that our models are compared against are discussed.

4.5.3.1.1. BLUEBERT model

The general language understanding evaluation(GLUE) benchmark has been long used for training, evaluating and analyzing natural language processing models. Using this as inspiration, BLUEBERT paper introduced the biomedical language understanding evaluation(BLUE) benchmark for NLP researchers in the biomedical field.

BLUE contains five tasks(sts, Name Entity Recognition, re, inference, and document classification) with pre-existing ten datasets that are widely used by the biomedical NLP research as shared tasks. To test the effectiveness of the BLUE benchmark, experiments were conducted on the BERT model. BLUEBERT is a BERT model pre-trained on PubMed abstracted and MIMIC III clinical notes. The model was able to achieve superior performance than SOTA models in many datasets especially those in the clinical domain.

For this reason, the BLUE benchmark was used to evaluate BioALBERT models, along with comparing our fine-tuning results to results of the BLUEBERT model.

4.5.3.1.2. BioBERT model

BioBERT is a domain-specific BERT model that was pre-trained on large biomedical corpus, Pubmed Abstracts and PMC full-text articles. In the BioBERT report, the BioBERT model was compared to original BERT which was pre-trained on only wiki + books corpus and SOTA models(each with different architecture). BioBERT was fine-tuned on three biomedical NLP tasks, Name Entity Recognition, RE and QA, with pre-existing fifteen widely used datasets for NLP biomedical research. The results show that the BERT model pre-trained on both PubMed abstracts and PMC achieves the best performance and is superior to the original BERT model. In some dataset, BioBERT was able to outperform SOTA models.

BioBERT is well-known and is one of the first BERT models to be used in the biomedical domain. Therefore, the BioBERT model is a good standard for comparison and used as one of our baseline. BioALBERT 2.0 & 2.1 models are fine-tuned on the same datasets as BioBERT NER and RE tasks.

4.5.3.2. Baseline Models for Vaccine-BERT fine-tuning

Similar to BioALBERT, the performance of Vaccine-BERT has to be tested against the baseline model. CT-BERT was chosen as our baseline model as it was also pre-trained on tweets corpus. In this section, this baseline model will be discussed in more detail.

4.5.3.2.1. CT-BERT

The CT-BERT model is a BERT based model trained on the corpus of tweets related to the COVID-19. Tweeter contains a large amount of unstructured expressions or messages that are hard to understand. To improve the understanding of contextual meaning of the sentences, CT-BERT introduces procedures to replace special tags such as emojis, URLs and username to common text token and ASCII representations before pretraining. After that, the performance of the model is evaluated on five sentiment classification tasks.

The Vaccine-BERT shared similar procedures on data pre-process and pretraining vaccine related tweets on BERT. It was fine-tuned on a vaccine related tweets dataset to evaluate the model performance. Therefore, CT-BERT was used as the baseline model on the same dataset. The impact of different pre-training on different domains are analyzed and compared.

4.5.4. Performance Evaluation

In this section, the evaluation techniques for BioALBERT and Vaccine-BERT are discussed.

4.5.4.1. BioALBERT Performance Evaluation

As mentioned in the previous section, BLUE benchmark was used to evaluate the performance of our model. BLUE contains a total of five different tasks. Each task has different metrics for evaluation.

The results are evaluated by comparing the pre-processed test set with the predicted results by the models. For NER datasets, token-level precision, recall and F1 score was used for evaluation. In RE tasks, the datasets used two different types of evaluation. DDI, ChemProt, and i2b2 used micro average precision, recall, and F1 score metrics, while the BioBERT RE datasets, GAD and euadr use weighted precision, recall, and F1 score. Sentence similarity datasets, both use pearson correlation coefficient to evaluate similarity. For inference tasks, accuracy is used to evaluate entails or contractions. Lastly, document classification uses F1 score for evaluation metric. Table below shows a summary table of BioALBERT evaluation metrics.

Corpus	Task	Used in	Metric
BC5CDR-Di	NER	BioBERT	F1
BC5CDR-Ch	NER	BioBERT	F1
NCBI Disease	NER	BioBERT	F1
JNLPBA	NER	BioBERT	F1
BC2GM	NER	BioBERT	F1
LINNAEUS	NER	BioBERT	F1
Species-800	NER	BioBERT	F1
GAD	RE	BioBERT	F1
euadr	RE	BioBERT	F1
ShARe/CLEFE	NER	BLUE	F1
DDI	RE	BLUE	Micro F1
ChemProt	RE	BLUE	Micro F1
i2b2 2010	RE	BLUE	Micro F1
HoC	Document Classification	BLUE	F1
MedNLI	Inference	BLUE	Accuracy
MedSTS	Sentence Similarity	BLUE	Pearson
BIOSSES	Sentence Similarity	BLUE	Pearson

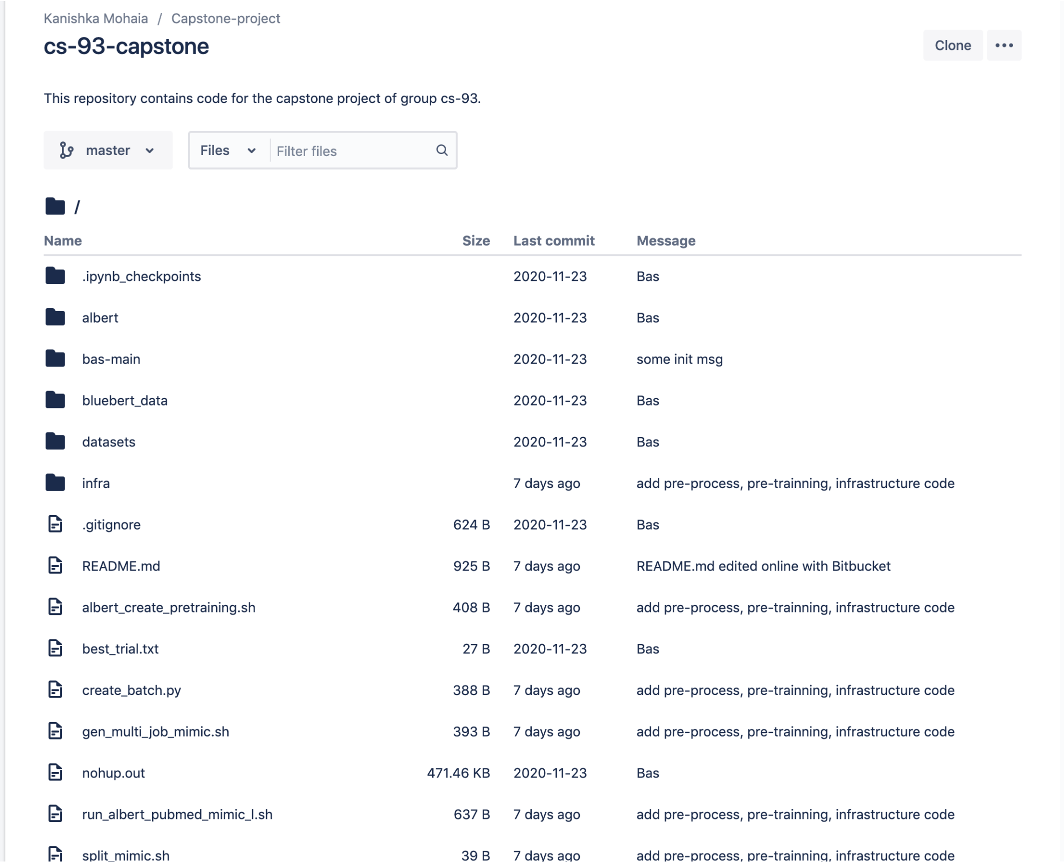
Table 21: Summary of BioALBERT performance evaluation[1][2]

4.5.4.2. Vaccine-BERT Performance Evaluation

The CT-BERT fine-tuning code i.e run_finetune.py will automatically output various evaluation metrics after finetuning has been completed.

5. Resources

In order to achieve the objectives of this project BERT and Albert were pre-trained and subsequently fine-tuned on downstream tasks. Pre-training as described in section 2.1.1 is a very resources intensive process hence for pre-training tasks google cloud TPU V3-8 were used. While fine-tuning was carried out a mix of GPUs and TPUs based on availability on google colab. The code for this project was written in Python using tensorflow 1.15 (Albert) and tensorflow 2.1 for CT-BERT. The code was stored on a private bitbucket repository.



Kanishka Mohaia / Capstone-project

cs-93-capstone

This repository contains code for the capstone project of group cs-93.

Name	Size	Last commit	Message
.ipynb_checkpoints		2020-11-23	Bas
albert		2020-11-23	Bas
bas-main		2020-11-23	some init msg
bluebert_data		2020-11-23	Bas
datasets		2020-11-23	Bas
infra		7 days ago	add pre-process, pre-trainning, infrastructure code
.gitignore	624 B	2020-11-23	Bas
README.md	925 B	7 days ago	README.md edited online with Bitbucket
albert_create_pretraining.sh	408 B	7 days ago	add pre-process, pre-trainning, infrastructure code
best_trial.txt	27 B	2020-11-23	Bas
create_batch.py	388 B	7 days ago	add pre-process, pre-trainning, infrastructure code
gen_multi_job_mimic.sh	393 B	7 days ago	add pre-process, pre-trainning, infrastructure code
nohup.out	471.46 KB	2020-11-23	Bas
run_albert_pubmed_mimic_l.sh	637 B	7 days ago	add pre-process, pre-trainning, infrastructure code
split_mimic.sh	39 B	7 days ago	add pre-process, pre-trainning, infrastructure code

Figure 30: bitbucket repository

5.1. Hardware & Software

The following table lists the hardware and software used in this project.

Task	Hardware	Software
Data analysis and pre-processing	Google BigQuery	SQL, Python, gcloud SDK
Pre-training of transformer base models	TPU V3 8 core instances on google cloud platform	gcloud SDK Tensorflow Enterprise 2.3 Python 3 Jupyter Notebook
Fine-tuning tasks	Google colab	gcloud SDK Tensorflow Enterprise 2.3 Python 3 Jupyter Notebook
Code repository	Bitbucket	Git
Team communication	NA	Slack/WhatsApp
Document and artifact storage	Google drive/goog storage buckets	gdrive

Table 22: Hardware & Software

5.2. Limitations

The main limitation that this project faced was of pre-training costs as cloud TPUs cost $8 USD/hr and $2.5 USD/hr for preemptible instances. Even though preemptible instances were used in this project the total cost of running pre-training ran up to $2000 AUD which was picked up by the team.

There were also team members who had large knowledge gaps which caused unnecessary pressure and bottlenecks in the team.

5.2.1. Limitation for Pre-train

The main limitation for pre-training was associated with its high costs, the team paid a little over $2000 to pre-train five different models and in order to investigate more models and perform more experiments in the time-frame more financial investment from team members was required.

5.2.2. Limitation for fine-tune

To save cost, all fine-tuning was done on Google colab. It has a maximum limit of TPU usage. As a countermeasure, we created multiple google accounts and switched to a new account every time the limit is reached. In total, 8 google accounts were used in the fine-tune process.

5.3. Materials

This section covered materials used. These materials were essential for project completion.

- **Pre-train BioALBERT model** - we initialise our models with BioALBERT weight trained on PubMed and PMC.
- **Original ALBERT model** - we use vocab and albert config files from albert github repository.

5.4. Roles & Responsibilities

In this section, roles and responsibilities of each team member is discussed.

Name	Roles & Responsibilities
Kanishka Mohaia	1. **Team leader** - Kanishka is the team leader. He organized the team meeting and assigned tasks according to team member's specialties. 2. **Pre-train** - Kanishka ran all the pre-train. This includes four BioALBERT models and Vaccine-BERT models. 3. **Report** - Organised and wrote the report.

Kasidej Cha-umpong	1. **Fine-tune** - Kasidej ran all the fine-tuning tasks for BioALBERT. He also ran fine-tuning for BioALBERT and the original ALBERT model to compare the results. 2. **Datasets** - Kasidej was responsible for finding all the datasets which are used for pre-train and fine-tune. He has to send applications and fill in the policy forms in order to get the access as most biomedical and clinical domain datasets are sensitive data of real people. 3. **Presentations and Report** - Kasidej helped Elroy organised the weekly presentation and wrote the final report.
Yu Him Elroy So	1. **Pre-process fine-tune datasets** - Elroy helped Kasidej with fine-tuning. Some datasets are raw files that need to be pre-process. He is responsible for this section. 2. **Presentations and reports** - Elroy was responsible for preparing weekly presentation slides and preparing final report writing.
Utkarsh Bele	1. **Literature review** - Utkarsh was tasked with researching all the Biomedical literature related to the project and carrying out a detailed literature review. **2. Report-** Utkarsh helped Elroy in writing the final report.
Siddhan Khondge	1. **Literature review** - Siddhan was tasked with carrying a detailed literature review on Biomedical language models.

Table 22: Roles & Responsibilities

6. Milestones

Beginning from week 2, our team started to study the previous work from the previous team and researched for any possible tasks to expand their model. In the first few weeks, our team set up communication channels on slack, trello and whatsapp to track study and work progress of each member. From week 2 to week 5, each team member was assigned a mission to look for any additional tasks from existing benchmark models and collect corresponding datasets. After handing out our proposal report on week 5, our team started implementing fine tuning tasks using available datasets at the moment. Due to access limits on some dataset like MedSTS and MIMIC and GPU quota limit in Google Cloud, the progress of fine tuning tasks was behind schedule. On week 9, our team managed to finish all new fine tuning tasks on BioALBERT and was waiting for the completion on pretraining our new BioALBERT model, meanwhile Kanishka finished pretraining vaccine tweets dataset on BERT model . A week later, upon the completion of the pretraining process on the ALBERT + MIMIC model, our team finalized all the fine tuning tasks from both BLUEBERT and BIOBERT on our new model and completed all tasks on week 13.

Milestone	Tasks	Reporting	Date
Week-1	Research on project related NLP language models	Client meeting to review the project requirement and project scope	05-09-2020
Week-2	Continues learning on BERT, ALBERT and BioALBERT	Client meeting to review the work plan	12-09-2020
Week-3	Design work plan	None	19-09-2020
Week-4	Gather datasets, clear data and create database	None	26-09-2020
Week-5	Proposal Report Due		03-10-2020
Week-6	Start fine tuning on existing datasets	Client meeting to review project progress	10-10-2020
Week-7	Fine tuning the models	None	17-10-2020
Week-8	Retraining and reviewing the results	Reporting the result of fine tuning tasks on BioALBERT	24-10-2020
Week-9	Progress Report Due		31-10-2020
Week-10	Completed all pre-training of 4 models with the MIMIC III dataset	Client meeting to deploy the result of fine tuning	07-11-2020
Week-11	Final Presentation		26-11-2020
Week-12	Final Report (thesis)		6-12-2020
Week-13	Documentation		13-12-2020

Table 23: Schedule & Milestones

The several major risks that we faced in this project including:

1. Long waiting time of access approval on some datasets
2. High computational cost on GPU and TPU consumption
3. Usage limit on Google Cloud GPU for fine tuning task
4. Knowledge gaps in team members
5. Technical issues with TF2.2.1 for pre-training

Major Risks		Likelihood (1-5)	Consequence (1-5)	Overall Risk (1-10)	Risk Level	Risk mitigation
R1	Long waiting time of access approval on some datasets	1	3	6	Low	Constant communication with the concerned authority and authors.
R2	High computational cost on GPU and TPU consumption	4	5	9	High	Used preemptible TPU to lower the costs.
R3	Usage limit on Google Cloud GPU for fine tuning task	4	4	9	High	Created multiple accounts to run fine tuning parallelly.
R4	Knowledge gaps in team members	4	4	8	High	Supervisor Mr.Usman assigned other tasks to those team members.
R5	Technical issues with TF2.2.1 for pre-training	4	4	8	High	Accepted the risk and ran the pretraining till 80K steps.

Table 24: Risks and Risk Mitigation

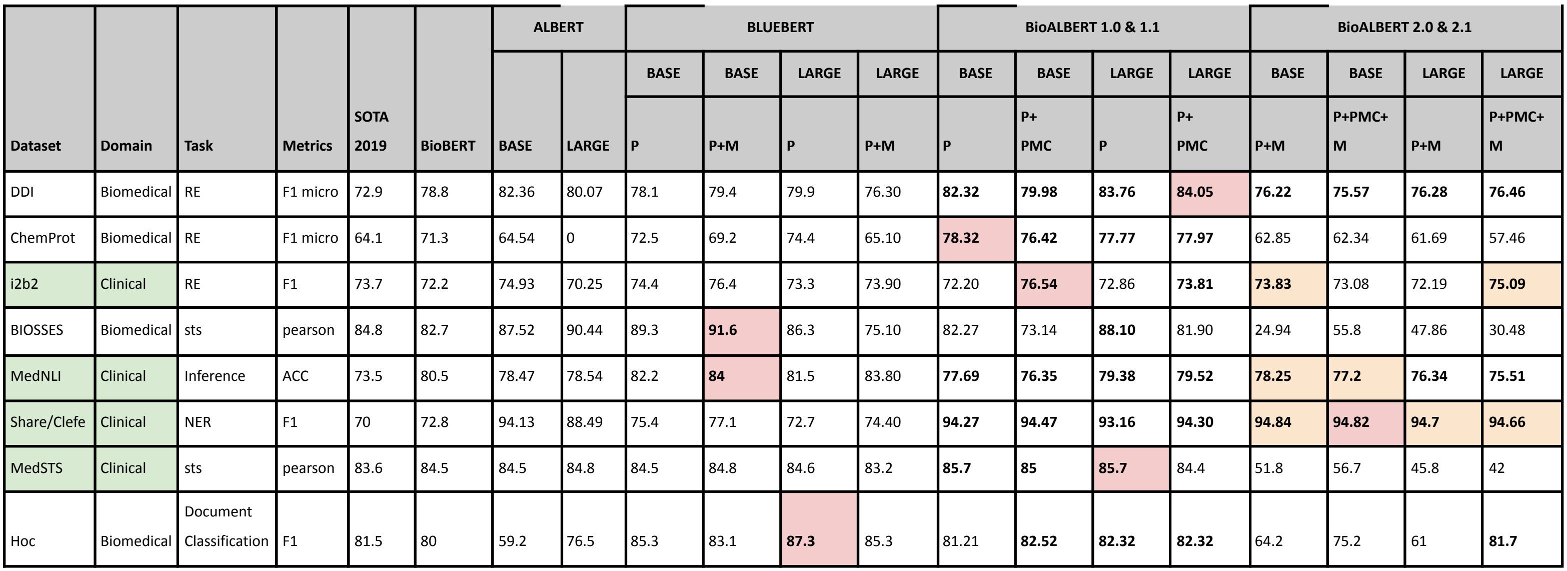
7. Results

7.1. BioALBERT Results

In this section, our fine-tuning results are presented. These results include BioALBERT, both the original 1.0 & 1.1 and our 2.0 & 2.1, on BLUE benchmark datasets and our BioALBERT fine-tuning results on BioBERT datasets. The original ALBERT was also fine-tuned to compare the effect of pre-training on specific domains.

In the figure below shows the fine-tuning results on the BLUE benchmark. Red cells represent the best model for each dataset. Orange cells are where BioALBERT 2.0 & 2.1 is superior to the original BioALBERT. Scores in bold text shows where our model outperforms SOTA models for each dataset.

BioALBERT models were able to outperform BLUEBERT in five out of eight datasets. The original ALBERT models achieved lower scores than pre-trained BioALBERT on all datasets, clearly showing the effect of pre-train. Comparing our BioALBERT 2.0 & 2.1 models with original BioALBERT, our models are superior to BioALBERT on datasets from the clinical domain. In the biomedical domain, the original BioALBERT models have excellent performance. BioALBERT achieved better performance than the BLUEBERT model on all RE datasets, ChemProt(+4.15%), i2b2(+5.82%), and DDI(+0.14%). On the other hand, the original ALBERT model achieved poor scores in RE tasks especially in the ChemProt dataset. ALBERT-LARGE was not able to find any relationship in the data. Our models also achieved relatively low scores on sentence similarity datasets. This can be due to mainly small data size, for example; BIOSSES only contains 20 samples in the test set. The uncertainty of the output Pearson coefficient is high.



						ALBERT		BLUEBERT				BioALBERT 1.0 & 1.1				BioALBERT 2.0 & 2.1			
								BASE	BASE	LARGE	LARGE	BASE	BASE	LARGE	LARGE	BASE	BASE	LARGE	LARGE
Dataset	Domain	Task	Metrics	SOTA 2019	BioBERT	BASE	LARGE	P	P+M	P	P+M	P	P+ PMC	P	P+ PMC	P+M	P+PMC+ M	P+M	P+PMC+ M
DDI	Biomedical	RE	F1 micro	72.9	78.8	82.36	80.07	78.1	79.4	79.9	76.30	**82.32**	**79.98**	**83.76**	**84.05**	**76.22**	**75.57**	**76.28**	**76.46**
ChemProt	Biomedical	RE	F1 micro	64.1	71.3	64.54	0	72.5	69.2	74.4	65.10	**78.32**	**76.42**	**77.77**	**77.97**	62.85	62.34	61.69	57.46
i2b2	Clinical	RE	F1	73.7	72.2	74.93	70.25	74.4	76.4	73.3	73.90	72.20	**76.54**	72.86	**73.81**	**73.83**	73.08	72.19	**75.09**
BIOSSES	Biomedical	sts	pearson	84.8	82.7	87.52	90.44	89.3	**91.6**	86.3	75.10	82.27	73.14	**88.10**	81.90	24.94	55.8	47.86	30.48
MedNLI	Clinical	Inference	ACC	73.5	80.5	78.47	78.54	82.2	**84**	81.5	83.80	**77.69**	**76.35**	**79.38**	**79.52**	**78.25**	**77.2**	**76.34**	**75.51**
Share/Clefe	Clinical	NER	F1	70	72.8	94.13	88.49	75.4	77.1	72.7	74.40	**94.27**	**94.47**	**93.16**	**94.30**	**94.84**	**94.82**	**94.7**	**94.66**
MedSTS	Clinical	sts	pearson	83.6	84.5	84.5	84.8	84.5	84.8	84.6	83.2	**85.7**	**85**	**85.7**	84.4	51.8	56.7	45.8	42
Hoc	Biomedical	Document Classification	F1	81.5	80	59.2	76.5	85.3	83.1	**87.3**	85.3	81.21	**82.52**	**82.32**	**82.32**	64.2	75.2	61	**81.7**

Table 25: BLUE benchmark fine-tuning results(Red cells: Best of all model, Orange cells: BioALBERT 2.0 & 2.1 outperform BioALBERT)

In the figure below shows the fine-tuning results on BioBERT datasets. Red cells represent the best model. Orange cells represent where out models, BioALBERT 2.0&2.1, outperform original ALBERT models.

Our BioALBERT 2.0&2.1 mostly outperform ALBERT in the BASE models. However, the performance difference is very slim, the difference is less than one percent. In NER tasks, our models were able to outperform BioBERT in some tasks, JNLPBA(+10%) and BC2GM(+0.32%). The original ALBERT outperformed BioBERT as much as three datasets, BC5CDR-disease, linnaeus, and s800. Despite that, BioBERT was still able to come out on top in two datasets, BC5CDR-chemical, and NCBI-disease. In RE tasks, BioBERT has superior performance than BioALBERT 2.0&2.1 and ALBERT models. The difference in performance is as much as 10% in both euadr and GAD datasets.

				ALBERT		BioBERT				BioALBERT 2.0 & 2.1			
						BASE		LARGE		BASE		LARGE	
Dataset	Domain	Task	Metrics	BASE	LARGE	P	PMC	P+PMC	P	P+M	P+PMC+M	P+M	P+PMC+M
BC5CDR-di	Disease	NER	F1	89.97	91.48	86.2	85.27	86.56	87.15	90.03	90.01	90.29	91.44
BC5CDR-Ch	Chemical	NER	F1	89.97	91.57	92.64	92.54	93.44	93.47	89.83	90.08	90.01	91.48
JNLPBA	Gene	NER	F1	85.73	76.23	76.65	76.53	77.59	77.49	86.74	86.56	86.2	85.72
linnaeus	Species	NER	F1	98.21	98.47	88.13	88.45	89.81	88.24	85.72	98.27	98.24	98.23
NCBI-di	Disease	NER	F1	86.16	87.64	87.38	87.79	89.36	89.71	85.82	85.93	85.86	85.83
s800	Species	NER	F1	93.81	93.63	73.08	73.09	75.31	74.06	93.53	93.63	93.63	93.63
BC2GM	Gene	NER	F1	83.24	84	82.54	83.53	84.40	84.72	83.35	83.38	83.44	84.72
euadr	Gene	RE	F1	66.78	77.81	84.44	86.51	84.83	79.74	62.52	76.93	70.41	70.48
GAD	Gene	RE	F1	70.71	68.60	81.61	80.24	81.52	79.83	72.68	69.14	71.81	68.17

Table 26: Fine-tuning results on BioBERT datasets(Red cells: Best of all model, Orange cells: BioALBERT 2.0&2.1 outperform ALBERT)

7.2. Vaccine-BERT Results

This section contains our fine-tuning results for vaccine-BERT. Table below compares fine-tuning Vaccine-BERT fine-tuning results with CT-BERT. From the table, CT-BERT performed an average of 10% better than our model.

Model	Accuracy	micro precision	micro recall	micro f1	matthews
CT-BERT	78.98	78.98	78.98	78.87	64.14
Vaccine-BERT	69.89	69.89	69.89	69.89	47.14

Table 27: Fine-tuning results of Vaccine-BERT

8. Discussion

The results of the project demonstrate that downstream tasks fine-tuned on the BioALBERT 2.0 and 2.1 outperform SOTA models on clinical datasets. In this project, additional corpus specifically MIMIC-III which a clinical dataset was used. The number of pre-training steps were also varied between 200K steps and 270K steps. It can be seen that the downstream tasks that belong to the clinical domain perform well and outperform SOTA models as shown by figure 33 & 34. The tasks shown are relation extraction, inference and named entity recognition.

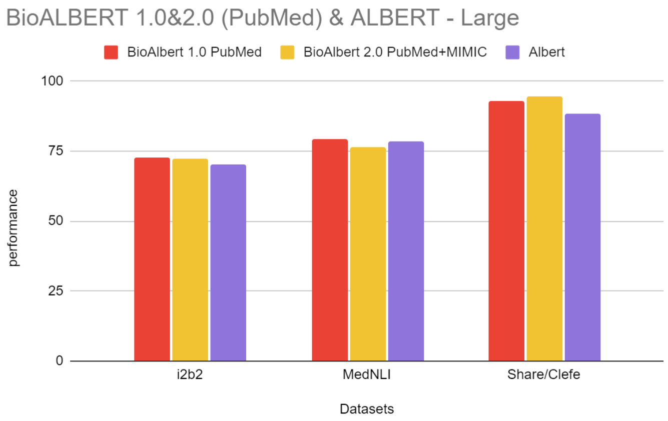


Figure 33: Performance bar chart of BioALBERT 1.0&2.0 and ALBERT LARGE models (Clinical datasets)

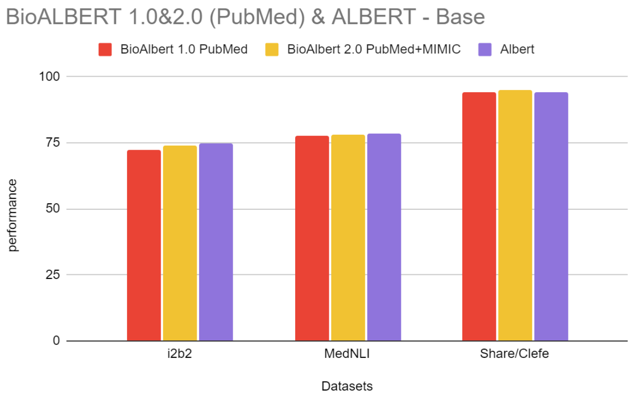


Figure 34: Performance bar chart of BioALBERT 1.0&2.0 and ALBERT BASE models (Clinical datasets)

It can be seen that the performance of the BioALBERT 2.0 improves as the corpus size increases on the three downstream tasks however since the corpus for MIMIC-III is 0.5 Billion words it has a minor impact when compared with English Wikipedia (2.5B), Books Corpus (0.8B), PubMed Abstracts (4.5B) and PMC (13.5 B). With PMC being the most dominant corpus size, it will have the largest impact on the performance of the models.

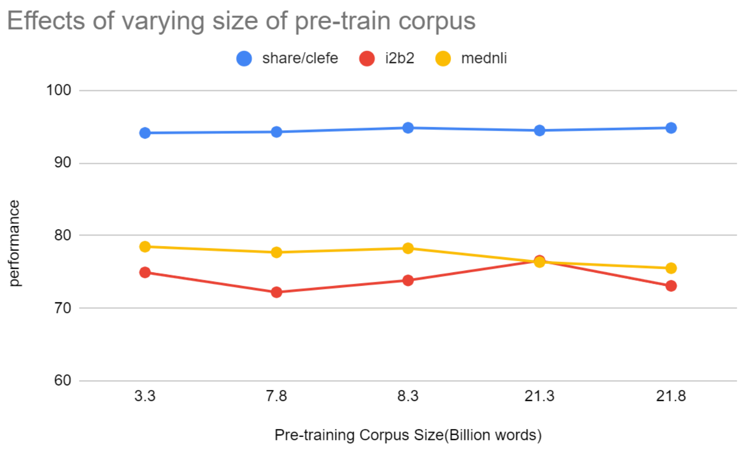


Figure 34 a. Showing performance change in relation to corpus size

The result of each pre-trained steps were also saved and a plot of how the number training steps impacts the accuracy of downstream tasks has also been analysed as shown by figure 34b. The following figure shows the change in performance with regards to training steps, it can be seen that as training steps increase the performance of downstream tasks improves.

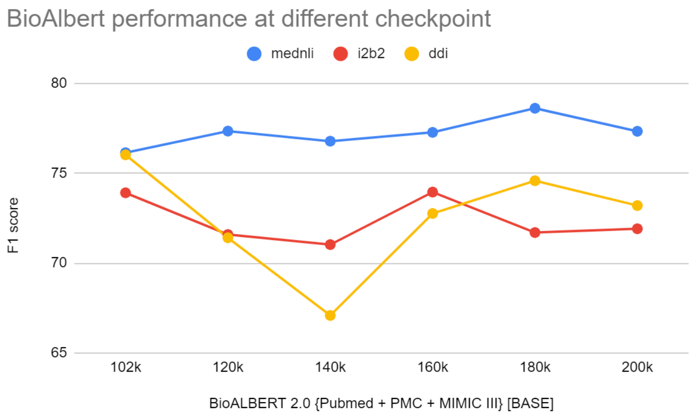


Figure 34b: Comparison of BioALBERT performance of 3 tasks on different checkpoints

Further predictions have been sampled for two different NER tasks, it can be seen that in almost all cases both BioAlbert 1.0 and BioAlbert 2.0 can recognise the named entities and identify the boundaries while plain Albert can not perform this task effectively. Hence Albert often produces incorrect results. This shows the effectiveness of BioALBERT 2.x on downstream tasks.

Prediction samples from BioALBERT 1.0&1.1, BioALBERT 2.0&2.1 and ALBERT on NER datasets			
Task	Dataset	Model	Sample
NER	JNLPBA	BioALBERT 2.0	Number of glucocoticoid **receptors** in lymphocytes and their sensitivity to...
		ALBERT	Number of glucocoticoid **receptors** in lymphocytes and their sensitivity to...
	Share/Clefe	BioALBERT 2.0	PATIENT/TEST INFORMATION: Indication: **Pericardial effusion**. Height:(in)...
		BioALBERT 1.0	PATIENT/TEST INFORMATION: Indication: **Pericardial effusion**. Height:(in)...
		ALBERT	PATIENT/TEST INFORMATION: Indication: Pericardial effusion. Height:(in)...
	Share/Clefe	BioALBERT 2.1	The **mitral** valve leaflets are mildly **thickened**. There is mild **mitral annular calcification**.TRICUSPID VALVE...
		BioALBERT 1.1	The **mitral** valve leaflets are mildly **thickened**. There is mild **mitral annular calcification**.TRICUSPID VALVE...
		ALBERT	The mitral valve leaflets are mildly thickened. There is mild mitral annular calcification.TRICUSPID VALVE...

Table 28: Prediction samples from of BioALBERT 1.0&2.0 and ALBERT Large models

BioAlbert 2.0 and 2.1 also perform well on non-clinical tasks and outperforming SOTA models indicating that it has wider utility than just being clinical specific. The following figures show the performance of BioAlbert 2.0 and 2.1 on data related to genes.

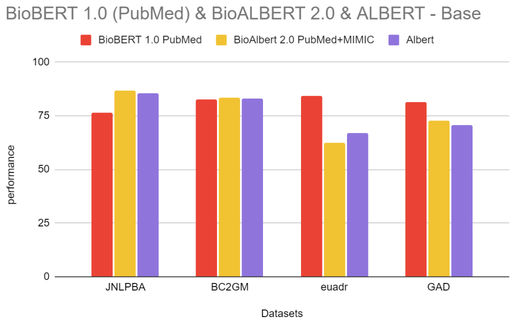


Figure 35: Performance bar chart of BioBERT 1.0 & BioALBERT 2.0 and ALBERT BASE models (Gene datasets)

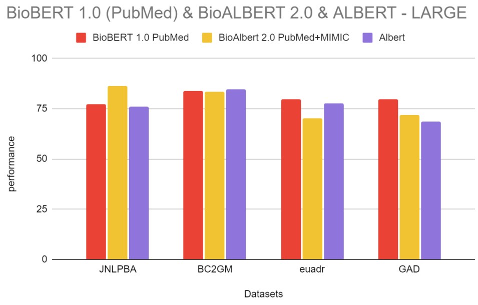


Figure 36: Performance bar chart of BioBERT 1.0 & BioALBERT 2.0 and ALBERT LARGE models (Gene datasets)

9. Conclusion

In this project BioALBERT 2.0 and 2.1 were trained on clinical datasets from the MIMIC-III dataset, it was shown that for some tasks specifically clinical datasets the model outperformed SOTA models. The models were also easily fine-tuned on downstream tasks without the need to make large scale changes to the underlying deep learning neural network architecture.

Furthermore this project has verified that generalised attention based architecture can be adopted on domain specific corpus and it can result in improved performance in this case clinical datasets. This project also provides opportunities for further research and has scope for future work where the model can be trained on PubMed, PMC and MIMIC datasets all together instead of using the pre-trained model weights as initialisations in order to increase the robustness and utility of the models on downstream tasks.

10. Limitations and Future Work

This section explains the limitations and some recommendations for future work that can be conducted by future teams to further improve this project.

10.1. Limitations

One of the main avenues for achieving novelty working with attention based models is to be able to perform many experiments and try new ideas with each iteration of the experiment. However, the main challenge is that pre-training BERT and its variants such ALBERT is very resources intensive which means in order to get a turn around in a reasonable amount of time which takes about ~4 days one would be paying $2.5 USD per hour if preemptible TPUs were used or $8 per hour if non-preemptible TPUs were used. This makes it very expensive for students if they were to pay this from their own pockets and as a result more experiments won't be performed.

Furthermore even if students are willing to spend money on google cloud TPUs they may not be able to get access to TPUs as google will limit the amount of resources that can be accessed by new users and especially if you are a student they will redirect you to your university or instruct you to use google COLAB which has very strong limits on usage. This can further cause bottlenecks and become impediments that can prevent students from running a wide variety of experiments and achieving novel results.

Finally other limitations include the unavailability of labeled and unlabelled data in the public domain. There are limited datasets that are being used by the research community and some of these datasets such as clinical notes data requires special access to be granted by the data owners which can be very time-consuming.

10.2. Future Work

Due to the limitations described in the previous section some of the work that the team planned to do could not be performed and can be potential projects for future teams which can further improve the results of these models.

A) **Creating a domain specific dictionary**: There is good reason to believe that creating a domain specific dictionary might help boost accuracy of the models. The team intended to create a variant of the BioALBERT 2.0 & 2.1 with domain specific dictionary however this would have cost the team another $2000. This can be attempted in the future by creating a domain specific dictionary at the pre-train stage. This has not been attempted yet as

BioBERT, BioALBERT 1.x and BioALBERT 2.x all use the default Bert and Albert dictionaries respectively.

B) **Pre-train BioALBERT 2.x with PMC, PubMed and MIMIC-III datasets** at the same time instead of initialising the model with BioALBERT 1.x weights. This should improve the accuracy of the models on more downstream tasks. This team was not able to perform this as it didn't have access to the pre-training data for BioALBERT 1.x.

C) **Hyper-parameter tuning of BioALBERT 2.x,** the hyper parameters specifically learning rate and embedding size used are the hyper parameters used by BioALBERT 1.x and BioBERT, there is room for exploring the hyperparameter space to help improve performance.

11. Acknowledgements

We would like to express our gratitude to Dr Matloob Khushi and our supervisor, Usman Naseem, for providing us necessary help on getting access to the MedSTS dataset from Mayo Clinic and vaccine tweets for model development. They also lead us to explore more possible solutions on model development. We also acknowledge the assistance from the BLUEBERT development team with thanks for responding to our enquiry on clarification of the BLUEBERT configuration. Last but not least, we thank the last team which developed the BioALBERT model provides us a well established model for this project. Without the help of them, this project would not succeed as expected.

12. References

1. BERT paper

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.[online] arXiv.org. Available at: https://arxiv.org/abs/1810.04805v1

2. ALBERT paper

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representation systems-23-31. [online] arXiv.org. Available at: https://arxiv.org/abs/1909.11942

3. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

Alammar, J., 2020. The Illustrated BERT, Elmo, And Co. (How NLP Cracked Transfer Learning). [online] Jalammar.github.io. Available at: <http://jalammar.github.io/illustrated-bert/> [Accessed 20 September 2020].

4. Attention Is All You Need

Vaswani, A., Polosukhin, I., Kaiser, L., N. Gomez, A., Jones, L., & Uszkoreit, J. et al. (2017). Attention Is All You Need. [online] arXiv.org. Available at: https://arxiv.org/pdf/1706.03762.pdf

5. Attention in Natural Language Processing

Galassi, A., Lippi, M., & Torroni, P. (2020). Attention in Natural Language Processing. [online] arXiv.org. Available at: https://arxiv.org/ftp/arxiv/papers/1902/1902.02181.pdf

6. BioBert:

Lee, J., Yoon, W., Kim, S., & Kim, D. et al. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. [online] arXiv.org. Available at: https://https://arxiv.org/ftp/arxiv/papers/1902/1902.02181.pdf

7. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets (BLUE)

Peng, Y., Yan, S., & Lu, Z. (2019). Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. [online] arXiv.org. Available at: https://arxiv.org/pdf/1906.05474.pdf

8. SciBert

Beltagy, I., Lo, K., & Cohan, A. (2020). SCIBERT: A Pretrained Language Model for Scientific Text. Available at: https://arxiv.org/pdf/1903.10676.pdf.

9. CT-BERT

Müller, M., Salathé, M. and Kummervold, P., 2020. COVID-TWITTER-BERT: A NATURAL LANGUAGE PROCESSING MODEL TO ANALYSE COVID-19 CONTENT ON TWITTER. [online] arXiv.org. Available at: <https://arxiv.org/pdf/2005.07503.pdf> .

10. BioALBERT: A Simple and Effective Pre-trained Language Model forBiomedical Named Entity Recognition

Naseem, U., Khushi, M., Reddy, V., Rajendran, S., Razzak, I. and Kim, J., 2020. BioALBERT: A Simple and EffectivePre-trained Language Model forBiomedical Named Entity Recognition. [online] Available at: <https://arxiv.org/pdf/2009.09223.pdf>

11. LINNAEUS: A species name identification system for biomedical literature

Gerner, M., Nenadic, G. and M Bergman, C., 2010. [online] Available at: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-85>

12. The Inner Works of BERT

McCormick, C., 2020. [online] Available at: <https://www.chrismccormick.ai/products/the-inner-workings-of-bert>

13. ALBERT vs. BERT Tutorial

McCormick, C., 2020. [online] Available at: <https://www.chrismccormick.ai/products/albert-ebook/categories/2221577/posts/7425499>

14. Word2Vec, SkipGram

Wang, Y., Liu, S., Afzal, N. and Rastegar-Mojarad, M., 2020. [online] Available at: <https://www.sciencedirect.com/science/article/pii/S1532046418301825>

15. ML-Net

Du, J., Chen, Q., Peng, Y. and Xiang, Y., 2020. [online] Available at: <https://arxiv.org/abs/1811.05475> [Accessed 7 November 2020].

16. Clinical BERT Embeddings

Alsentzer, E., Murphy, J., Naumann, T. and Jin, D., 2020. [online] Available at: <https://arxiv.org/abs/1904.03323>

17. Automatic semantic classification of scientific Literature according to HOC

Baker, S., Silins, I., Korhonen, A. and Stenius, U., 2016. [online] Available at: <https://pubmed.ncbi.nlm.nih.gov/26454282/> [Accessed 7 November 2020].

18. Text Mining

Peng, J. and Sun, H., 2014. [online] Available at: <https://www.researchgate.net/publication/286789466_Preprocessing_in_Biomedical_Literature_Mining_Using_Natural_Language_Processing> [Accessed 7 November 2020].

19. BiO-Net

Xiang, T., Zhang, C. and Liu, D., 2020. [online] Available at: <https://arxiv.org/abs/2007.00243> [Accessed 7 November 2020].

20. NCBI disease corpus: a resource for disease name recognition and concept normalization

Islamaj Doğan, R., Leaman, R. and Lu, Z., 2014. [online] Available at: <https://pubmed.ncbi.nlm.nih.gov/24393765/> [Accessed 21 November 2020].

21. Bc2gm

Smith, L. and K Tanabe, L., 2008. [online] Available at: <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2008-9-s2-s2#Abs1> [Accessed 21 November 2020].

22. ShAReCLEF eHealth 2013: Natural Language Processing and Information Retrieval for Clinical Care v1.0

Mowery, D., 2013. [online] Available at: <https://physionet.org/content/shareclefehealth2013/1.0/> [Accessed 21 November 2020].

23. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text

Uzuner, Ö. and L DuVall, S., 2010. [online] Available at: <https://academic.oup.com/jamia/article/18/5/552/830538>

24. Chemprot

Arighi, C., Wang, Q. and Wu, C., 2017. [online] Available at: <https://biocreative.bioinformatics.udel.edu/media/store/files/2017/ProceedingsBCVI_v2.pdf?page=141>

25. DDI

Martinez, P., 2013. [online] Available at:
<https://www.sciencedirect.com/science/article/pii/S1532046413001123?via%3Dihub>

26. Visualizing A Neural Machine Translation Model (Mechanics Of Seq2seq Models With Attention)

Alammar, J., 2020. Visualizing A Neural Machine Translation Model (Mechanics Of Seq2seq Models With Attention). [online] Available at:
<https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/>

27. The Illustrated Transformer

Alammar, J., 2018. *The Illustrated Transformer*. [online] Available at:
<http://jalammar.github.io/illustrated-transformer/>

28. CBOW and Skip-grams

Chiu, B., Crichton, G., Korhonen, A. and Pyysalo, S., 2016. [online] Available at:
<https://www.aclweb.org/anthology/W16-2922/>

29. Bag of Words (BOW), InferSent and ESIM

Romanov, A. and Shivade, C., n.d. [online] Available at:
<https://www.aclweb.org/anthology/D18-1187.pdf>

30. CollaboNet: collaboration of deep neural networks for BioNER

Yoon, W., Ho So, C., Lee, J. and Kang, J., 2020. [online] Available at:
<https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2813-6>

31. Machine learning for concept extraction on clinical documents from multiple data sources

Torii, M., Wagholikar, K. and Liu, H., 2020. [online] Available at:
<https://pubmed.ncbi.nlm.nih.gov/21709161/>

32. BioSentVec: creating sentence embeddings for biomedical texts

Chen, Q., Peng, Y. and Lu, Z., 2018. BioSentVec: creating sentence embeddings for biomedical texts. [online] Available at: <https://arxiv.org/ftp/arxiv/papers/1810/1810.09302.pdf>

33. Enhancing Clinical Concept Extraction with Contextual Embeddings

Si, Y., Wang, J., Xu, H. and Roberts, K., 2019. Enhancing Clinical Concept Extraction with Contextual Embeddings. [online] Available at: <https://arxiv.org/pdf/1902.08691.pdf>

34. Comparative study of various machine learning classifiers on medical data

N. Karankar, P. Shukla and N. Agrawal,2017. Comparative study of various machine learning classifiers on medical data.[online] Available at: <https://ieeexplore.ieee.org/document/8418550>

35. Probing Biomedical Embeddings from Language Models

Jin, Q., Dhingra, B., Cohen, W. and Lu, X., 2019. Probing Biomedical Embeddings from Language Models. [online] Available at: <https://www.aclweb.org/anthology/W19-2011.pdf>

36. Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray

Rahman, T., Chowdhury, M., Khandakar, A., Islam F, K., Mahbub, Z., Kadir, M., Kashem, S. and Islam R, K., 2020. Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray. [online] Available at: <https://arxiv.org/ftp/arxiv/papers/2004/2004.06578.pdf>

37. Challenges in clinical natural language processing for automated disorder normalization

Leaman, R., Khare, R., & Lu, Z. (2015). Challenges in clinical natural language processing for automated disorder normalization. Journal Of Biomedical Informatics,[online] Available at: <https://www.researchgate.net/publication/280116631_Challenges_in_Clinical_Natural_Language_Processing_for_Automated_Disorder_Normalization>

38. Characterizing Subgroups of High-Need, High-Cost Patients Based on Their Clinical Conditions: a Machine Learning-Based Analysis of Medicaid Claims Data

Nuti, S., Doupe, P., Villanueva, B., Scarpa, J., Bruzelius, E. and Baum, A., 2019. Characterizing Subgroups of High-Need, High-Cost Patients Based on Their Clinical Conditions: a Machine Learning-Based Analysis of Medicaid Claims Data. *Journal of General Internal Medicine* [online] Available at:<https://link.springer.com/article/10.1007/s11606-019-04941-8>

39. Adversarial Training for Relation Extraction

Wu, Y., Bamman, D. and Russell, S., n.d. Adversarial Training for Relation Extraction, [online] Available at: <https://www.aclweb.org/anthology/D17-1187.pdf>

40. Machine learning in the analysis of medical data

Kazakov, O., Averchenkov, A. and Kulagina, N., 2018. Machine learning in the analysis of medical data. [online] Available at: <https://iopscience.iop.org/article/10.1088/1742-6596/1210/1/012061/pdf>

41. Controlling testing volume for respiratory viruses using machine learning and text mining

V, M. and K, M., n.d. Controlling testing volume for respiratory viruses using machine learning and text mining. [online] Available at: <https://pubmed.ncbi.nlm.nih.gov/28269950/>

42. MIMIC

Johnson, A., Pollard, T,J., Shen, L., Lehman, L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, LA., and Mark, R, G. MIMIC-III, a freely accessible critical care database (2016).[online] Available at: <https://mimic.physionet.org/>

43. Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework

Sanad, M., 2019. *Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework*. [online] Available at: <https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/>

13. Appendix

13.1. Appendix 1

13.1.1. Literature Summary Table

Model/ Method/Name	Architecture	Pre Training Corpus	Fine Tuning Corpus	Description	Limitation/ Drawback	Results
BERT	Base model 12 Layers with 108M parameters and Large with 24 layers and 334M parameters.	Books and English Wikipedia corpus	MNLI-(m/mm) QQP QNLI SST-2 CoLA STS-B MRPC RTE	It pretrains bidirectional representations on unlabeled text on both left and right context	Computationally expensive.	Dev set result for BERT base average at 82.3 and Large at 85.2
ALBERT	Base with 12 layers with 12M parameters and Large model 24 layers with 18M parameters	Books and English Wikipedia corpus	SQuAD1.1, SQuAD2.0, MNLI SST-2 RACE	Uses parameter reduction techniques which leads to 18x fewer parameters compared to BERT large and 1.7x faster.	ALBERT XXlarge less parameters compared to BERT large but computationally more expensive.	Dev set result for ALBERT XXlarge performs best with 88.7%.
BioBERT	Based on BERT architecture	English Wikipedia, Books corpus, PubMed Abstracts and PMC full text articles.	NCBI Disease ,i2b2/VA ,BC5CDR Disease,BC5CDR Chemical ,BC4CHEMD ,BC2GM ,JNLPBA	A model based on BERT which is pretrained on Bio medical corpora.	Computationally expensive and require large pre training time	BioBERT achieved 0.62 micro F1 in NER task than SOTA, RE 2.8 higher than SOTA, for QA MRR score 7 higher than SOTA.

			,LINNAEUS, Species-800 ,GAD,EU-ADR ,CHEMPROT, BioASQ 4b-factoid,BioASQ 5b-factoid,BioASQ 6b-factoid			
SciBERT	Based on BERT architecture	Books and English Wikipedia corpus, Random sample 1.14M papers on Semantic scholars	BC5CDR, JNLPBA, NCBI-disease, EBM-NLP, GENIA(LAS and UAS), ChemProt,SciERC, SciERC, ACL-ARC, Paper Field SciCite	It is a pre-trained language model based on BERT but trained on a large corpus of scientific text to improve performance on the downstream tasks by using its own in domain vocabulary	Model is computationally expensive to train.	SciBERT outperforms BERT Base on average by 2 percent.
BLUEBERT	Based on BERT architecture pre trained on medical corpus	Books and English Wikipedia corpus, MIMIC III and PubMed abstracts	MEDSTS, BIOSSES, BC5CDR, SHARE/CLEFE,DDI, I2b2, Chemprot, HOC,MEDNLI.	A benchmark created for the development of pretraing language representations in Biomedical corpus	Model is computationally expensive.	Overall BLUEBERT BASE achieved best results on 5 tasks than SOTA.

CT-BERT	Based on BERT Large(uncased, whole word masking) architecture	160M Covid tweets from Twitter from Jan 12 to April 16 2020	COVID-19 Category,Vaccine Sentiment, Maternal Vaccine Stance (MVS),Twitter Sentiment SemEval, Stanford Sentiment Treebank 2 (SST-2)	Model pretrained on Covid tweets for the analysis of this corpora.	Used optimization of BERT instead of in-domain optimization of CT-BERT	COVID-19-related dataset had the highest improvement with average 3.3% improvement on all datasets.
BioSentVec	Sentence Encoder/Embeddings pretrained on PubMed and MIMIC III Clinical notes	Pubmed, MIMIC III Clinical Notes	BIOSSES and MedSTS	Sentence encoder pretrained on Biomedical text corpus over 30 million documents for sentence similarity task	Can perform better on only sentence similarity tasks.	highest performance was obtained based on the proposed BioSentVec embeddings in both supervised and unsupervised methods as compared with the other approaches such as averaged word vectors or sentence vectors trained from the general domain. In unsupervised method

						BioSentVec (PubMed) Achieved a score of 81.7. in supervised method Deep learning + BioSentVec (PubMed + MIMIC-III) outperformed with 84.8
Different Word Level and Contextual Embeddings	word2vec, GloVe, fastText, BERT(BASE and LARGE) , ELMO and BioBERT	Gigaword5 + Wikipedia2014,Wikipedia 2017+ UMBC, WMT 2008-2012 + Wikipedia,BooksCorpus+ English Wikipedia,MIMIC III	I2B2 2010, I2B2 2012, SemEval 2014 Task 7 and SemEval 2015 Task 14	Different Word Embeddings methods evaluated using the general model and pretrained on the MIMIC III model for concept extraction tasks.	Long pre training time required for BERT and ELMO and computationally expensive.	For i2b2 2010, the best performance is achieved by BERTLARGE(MIMIC) with an F-measure of 90.25. The best performance on the i2b2 2012 task is achieved by BERTLARGE(MIMIC) with an F- measure of 80.91 The most efficient model for SemEval 2014 task achieved an exact

						matching with F-measure of 80.74 by BERTLARGE(MIMIC).
Machine Learning Classifiers on Medical Data	Classifiers used KNN classifier, Naive Bayes classifier, Support Vector Machine, Neural Network, Gaussian Mixture Model and Decision Tree classifier.		UCL dataset used which consist of data of 303 patients	Different Classifiers used for the Data Classification task on a single dataset.	Dataset used was smaller in size.	Naive Bayes outperforms all the other classifiers with 58.49.
Biomedical Embeddings from Language Models	BioELMo,BioBERT,General ELMo,General BERT,Biomed w2v	----	SNLI dataset,MEDNLI,PC2GM,CONLL 2003	They designed two tasks for probing which are NER,NLI which they used on the two models BioBERT and BioELMo for fine-tuning to find which model performs well in the probing task.	Pretraining on domain specific information datasets may lose general control settings	BioELMo and BioBERT representations are highly effective on biomedical NER and NLI, and BioELMo works even without complicated downstream models and outperforms

						untuned BioBERT in probing tasks BioElmo the score obtained was 57.5 and followed by ELMO with 53.3
AlexNet, ResNet18, DenseNet201 and SqueezeNet	Pretrained Convolutional Neural Network (CNN) models		Kaggle chest X-ray pneumonia database with 5247 chest x-ray images.	Transfer learning on 4 pre trained CNNs for Classification task.	CNN requires large dataset for better performanc e	The classification accuracy, precision and recall of normal and pneumonia images, bacterial and viral pneumonia images, and normal, bacterial and viral pneumonia were (98%, 97%, and 99%); (95%, 95% and 96%) and (93.3%, 93.7% and 93.2%) respectively.
CBOW and Skip-grams	Word Vectors	PubMed, PMC and PubMed+PM C		Performance of Word vectors with respect to different corpora with different	Number of training iterations was not taken into	Performance of word vectors changes with different corpora.

				preprocessing options and different hyper parameters settings.	consideration.	
Bag of Words (BOW), InferSent and ESIM	Deep neural networks	MIMIC III v1.3	MedNLI.	Created MedNLi dataset for natural language inference in the clinical domain.	The MedNLI dataset is single annotated.	InferSent performs best in dev(76%) and test(73.5%) set for MedNLI.
CollaboNet: collaboration of deep neural networks for BioNER	Bidirectional LSTM with CRF(Conditional Random Field)		BC2GM, BC4CHEMD BC5CDR,JNLPBA,NCBI constructed from MEDLINE abstracts	Combination of multiple NER models used on the NER datasets to imp	As the NER datasets are scarce resources. Thus Need more datasets related to NER to implemented model to achieve successfully reduced the number of misclassified entities and improved the performance	CollaboNet has improved both precision and recall. CollaboNet also outperforms the multi-task model (MTM) And multi-task learning has improved performance in previous studies
Machine learning for concept extraction on clinical documents	BioTagger-GM to train machine learning taggers	BETH, PARTNERS, UPMCD, UPMCP		Trained taggers were evaluated using the annotated clinical documents	additional training corpus would improve the machine	F scores range from 0.347 to 0.651. Thus F scores, ranging from 0.787 on BETH for test

from multiple data sources				made available in the 2010 i2b2/VA from the four sources	learning taggers and performanc e. As the data is limited in the clinical field.	concepts to 0.890 on UPMCP the performance was improved with the help of additional datasets for training data.BETH and UPMCP and that trained on the combination of BETH, PARTNERS, and UPMCP achieved F scores of 0.882 and 0.888, respectively .
Word2Vec, SkipGram	Word Embedding	EHR, GloVe, MedLit, Google News	I2b2 2006, DDI 2013, Pedersens, Hliaoutakis, MayoSRS, UMNSRS, infNDCG, TrREC 2016, DrugBank, MedLine and both combined	Compared word embeddings trained on different biomedical text corpus.	used only two word embeddings	semantic similarity captured by the word em-beddings trained on EHR are closer to human experts, i2b2-EHR trained word embedding had 0.9 F1 score, TREC-map score of

						0.067 on baseline word embedding, DDI- combined dataset performed best with F1 score of 0.79 on word embedding trained on google news corpus.
ML-Net (multi-label classification of biomedical texts with deep neural networks)	There major modules 1) Number of tokens in sentence 2)Number of sentences in document 3)Number of labels in document	PubMed word2vec (200) MIMIC III word2vec (300)	Hallmarks of cancers Classification, Chemical exposure Assessments, Diagnosis codes assignment	Evaluated ML-net with the three text classification tasks with two publicly available datasets in biomedical literature and clinical notes Types of text genres.	Current models are not suitable to perform on large scale multi-label biomedical text	In the task 1 and task 2 the F1 score has outperformed if we see Task 1: 0.714 Task 2: 0.724 As compared to the task 3: 0.395
Clinical BERT Embeddings	Base model 12 Layers with 108M parameters and Large with 24 layers and 334M parameters.	Books and English Wikipedia corpus, PubMed Abstracts and PMC full text articles, MIMIC III(Clinical notes and Discharge Summaries)	MedNLI and I2B2 2006 ,2010,2012 & 2014	Trained BERT base and BioBERT models on clinical notes and discharge summaries for different tasks.	Only used MIMIC(source single institution) for training dn did not used advance models	BioBERT performed best in i2b2 2006 with 94.8% accuracy, BioBERT trained on clinical notes and discharge summaries with 82.7%

						accuracy on MedNLI
Automatic semantic classification of scientific Literature according to HOC	HOC classifiers (PS,GS,CD,RI,A,IM,GI,TPI,CE,ID)	Hallmarks of Cancer (HOC)	----	LIBSVM classifier was used for the experiment. They also experimented on linear kernel and non-linear kernel.	overlapping capabilities (e.g. cell growth) and is likely to be the main reason for the lower classifier performance	results for each of the 10 trained hallmark classifiers. The accuracy of the 10 hall mark classifiers is impressive ranging between 83.4% and 98.9%. Average F1 score was 76.9% three categories scoring above 80% apart from other six category
Text Mining	Biomedical Literature Mining		1000 sentences from Bilingual text in English Readings	They have tried to find the precision of Word count, Corpus count and combination using 4 different contexts in order to reduce noise in data.	Their test set is very small for concrete results.	They achieved precision of 89.1% on all article error classes(combination) and 88.7% loss error class

BiO-Net	Recurrent Bi-directional Connections for Encoder-Decoder Architecture		MoNuSeg, TNBC, EM, MICCAI15 CBTC.	Compared different U-net variants and BiO-Net on three different medical imaging tasks	Has reused parameters of U-net.	BiO-Net significantly outperforms U-net.
DNorm-C	linear-chain conditional random fields	ShARe/CLEF eHealth Task 1 Corpus, NCBI Disease		DNorm-C is a high performing, open source system for finding disorders in clinical text	Manually analysed the errors, Couldn't identify disorders that the Annotator did	ShARe/CLEF eHealth Task. For NER (strict span-only), their system achieves precision = 0.797,recall = 0.713, f-score = 0.753. For the normalization task (strict span + concept) it achieves precision = 0.712, recall = 0.637, f-score = 0.672.
Machine Learning Concepts	Decision tree classifier algorithm and Data mining	EMR data of 1,685 pediatric inpatients receiving respiratory virus testing from 2010-2012.		Created a model to predict the outcome of laboratory tests for individual respiratory viruses.	Sample size is small, Variable writing style in notes	Basic model showed some predictive ability (AUC > 0.65) for four of the six viruses under study: influenza, parainfluenza, RSV, and hMPV.

Machine learning model	Classification learning algorithm - Stochastic Gradient Descent.	1102 General Blood Tests	Developed a model that will diagnose certain chronic obstructive pulmonary diseases or allergic rhinitis with a sufficient degree of probability.	Sample size is small	Reached an accuracy of 0.87 for of diagnosis of diseases.
Machine learning Method	Affinity propagation (AP)- Clustering Algorithm	Administrative claims from 34,764 patients insured by Medicaid between 1/1/2014 and 12/31/2015.	Describe different subgroups of HNHC patients based on their clinical characteristics for an urban Medicaid population in the Mount Sinai Health System (MSHS).	Large variation in average costs, used open source method	largest HNHC patient subgroups were characterized by mental and behavioral health conditions.
CNN and RNN	Adversarial training on RE (is mean of regularizing classification algorithms)	NYT and UW dataset	Demonstrate that adversarial training is generally effective for both CNN and RNN models and significantly improves the precision of predicted relations.	adversarial training works most effectively when only producing tiny perturbations on word features while keeping the semantics of sentences unchanged	RNN generally produces more precise predictions than CNN

13.2.

13.3. Appendix 2

Fine-tuning results precision, recall, and f1 score on the following models

- BioALBERT 2.0 (+ PubMed + MIMIC III 200K) [Base]
- BioALBERT 2.0 (+ PubMed + PMC + MIMIC III 200K) [Base]
- BioALBERT 2.1 (+ PubMed + MIMIC III 270K) [Large]
- BioALBERT 2.1 (+ PubMed + PMC + MIMIC III 270K) [Large]

ChemProt	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.91	0.45	0.60
BioALBERT 2.0 (P + PMC + M) [Base]	0.90	0.48	0.62
BioALBERT 2.1 (P + M) [Large]	0.89	0.47	0.62
BioALBERT 2.1 (P + PMC + M) [Large]	0.89	0.42	0.57

i2b2	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.92	0.63	0.75
BioALBERT 2.0 (P + PMC + M) [Base]	0.91	0.61	0.73
BioALBERT 2.1 (P + M) [Large]	0.91	0.62	0.74
BioALBERT 2.1 (P + PMC + M) [Large]	0.89	0.65	0.75

ddi	precision	recall	f1
BioALBERT 2.0 (P + M) [Base]	0.90	0.66	0.76
BioALBERT 2.0 (P + PMC + M) [Base]	0.91	0.65	0.76
BioALBERT 2.1 (P + M) [Large]	0.90	0.66	0.76
BioALBERT 2.1 (P + PMC + M) [Large]	0.90	0.67	0.76

MedNLI	precision	recall	f1
BioALBERT 2.0 (P + M) [Base]	0.78	0.78	0.78
BioALBERT 2.0 (P + PMC + M) [Base]	0.77	0.77	0.77
BioALBERT 2.1 (P + M) [Large]	0.77	0.77	0.77
BioALBERT 2.1 (P + PMC + M) [Large]	0.75	0.75	0.75

Share/Clefe	precision	recall	f1
BioALBERT 2.0 (P + M) [Base]	0.95	0.95	0.95
BioALBERT 2.0 (P + PMC + M) [Base]	0.95	0.95	0.95
BioALBERT 2.1 (P + M) [Large]	0.95	0.95	0.95
BioALBERT 2.1	0.95	0.95	0.95

(P + PMC + M) [Large]			

BC5CDR-chem	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.89	0.90	0.90
BioALBERT 2.0 (P + PMC + M) [Base]	0.89	0.91	0.90
BioALBERT 2.1 (P + M) [Large]	0.89	0.91	0.90
BioALBERT 2.1 (P + PMC + M) [Large]	0.89	0.94	0.91

BC5CDR-disease	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.89	0.91	0.90
BioALBERT 2.0 (P + PMC + M) [Base]	0.89	0.91	0.90
BioALBERT 2.1 (P + M) [Large]	0.89	0.92	0.90
BioALBERT 2.1 (P + PMC + M) [Large]	0.89	0.94	0.91

euadr	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.54	0.74	0.63
BioALBERT 2.0 (P + PMC + M) [Base]	0.77	0.77	0.77
BioALBERT 2.1	0.75	0.77	0.73

(P + M) [Large]			
BioALBERT 2.1 (P + PMC + M) [Large]	0.71	0.70	0.70

GAD	precision	recall	f1
BioALBERT 2.0 (P + M) [Base]	0.73	0.73	0.73
BioALBERT 2.0 (P + PMC + M) [Base]	0.69	0.69	0.69
BioALBERT 2.1 (P + M) [Large]	0.72	0.72	0.72
BioALBERT 2.1 (P + PMC + M) [Large]	0.72	0.69	0.68

JNLPBA	precision	recall	f1
BioALBERT 2.0 (P + M) [Base]	0.88	0.86	0.87
BioALBERT 2.0 (P + PMC + M) [Base]	0.87	0.86	0.87
BioALBERT 2.1 (P + M) [Large]	0.87	0.86	0.86
BioALBERT 2.1 (P + PMC + M) [Large]	0.86	0.86	0.86

linnaeus	precision	recall	f1
BioALBERT 2.0 (P + M) [Base]	0.98	0.99	0.98
BioALBERT 2.0	0.98	0.99	0.98

(P + PMC + M) [Base]			
BioALBERT 2.1 (P + M) [Large]	0.98	0.99	0.98
BioALBERT 2.1 (P + PMC + M) [Large]	0.98	0.98	0.98

NCBI-diease	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.85	0.87	0.86
BioALBERT 2.0 (P + PMC + M) [Base]	0.85	0.87	0.86
BioALBERT 2.1 (P + M) [Large]	0.85	0.87	0.86
BioALBERT 2.1 (P + PMC + M) [Large]	0.85	0.87	0.86

s800	**precision**	**recall**	**f1**
BioALBERT 2.0 (P + M) [Base]	0.92	0.95	0.94
BioALBERT 2.0 (P + PMC + M) [Base]	0.92	0.96	0.94
BioALBERT 2.1 (P + M) [Large]	0.92	0.96	0.94
BioALBERT 2.1 (P + PMC + M) [Large]	0.92	0.96	0.94

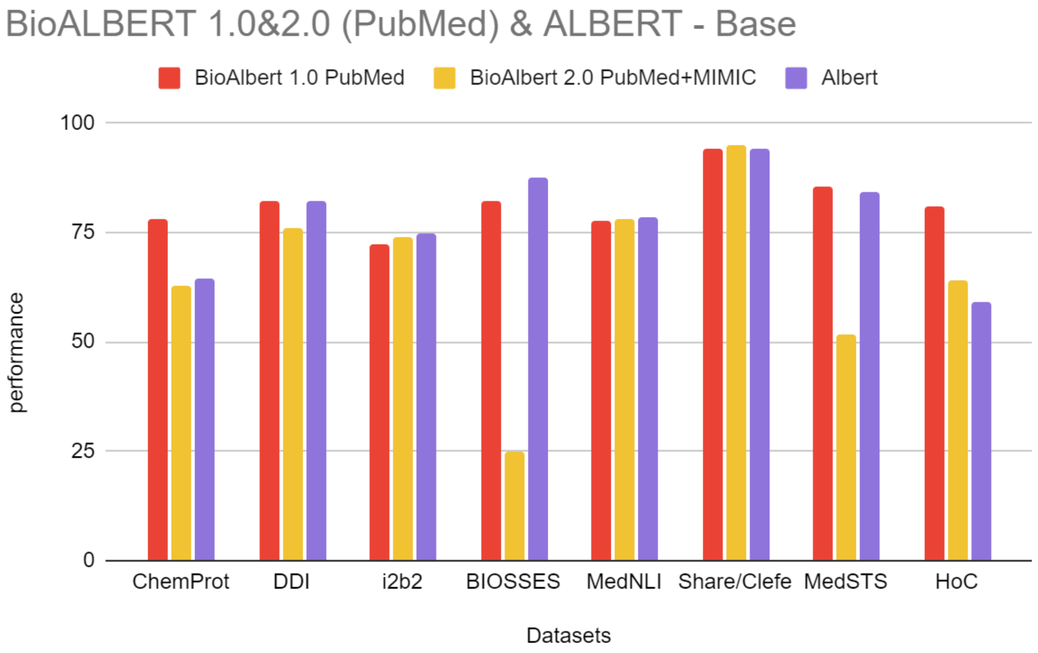


Figure 31: Performance bar chart of BioALBERT 1.0&2.0 and ALBERT BASE models

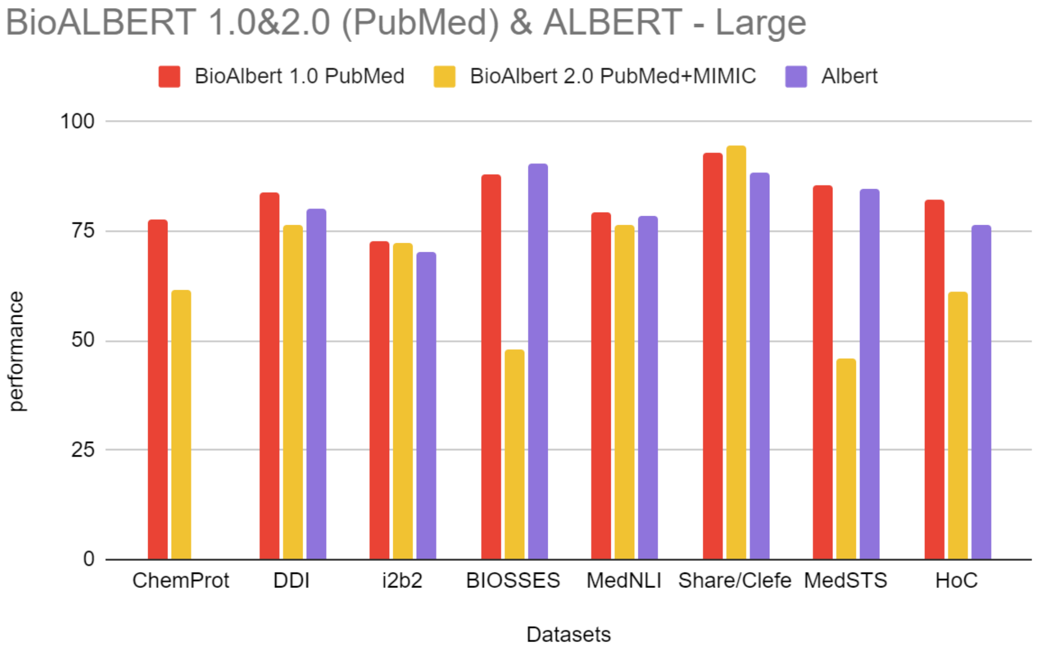


Figure 32: Performance bar chart of BioALBERT 1.0&2.0 and ALBERT BASE models