Comparative Analysis of Fine-tuning Pre trained Models on Domain Specific Dataset

Hitesh Patel
New York University
hlp276@nyu.edu

I. Introduction:

Question Answering is a computer science discipline within the fields of information retrieval and natural language processing, which focuses on building systems that automatically answer questions posed by humans in a natural language. One of the biggest challenges in NLP is the lack of enough training data. Overall, there is enormous amount of text data available, but if we want to create task-specific datasets, we need to split that pile into the very many diverse fields. Deep learning models tend to perform exceptionally well when they are trained on millions, or billions, of annotated training examples. To solve this problem researchers at google came up with pretrained model like **BERT (Bidirectional Encoder Representations from Transformer)** using various techniques for training general purpose language representation models using the enormous amount of unannotated text on the web. Bert is trained on corpus of Wikipedia and books. These general purpose pre-trained models can then be fine-tuned on smaller task-specific datasets. SquAD dataset has been used for question answering task by finetuning on BERT and has given state of the art results. It consists of title, paragraph which contains context, Qas. Qas contains information about answers, answers start and id. In our study we want to examine if this pre trained models which are trained on general corpus do really perform well or do they need to be trained on task specific dataset. In this project we are using pretrained general language models trained on large amount of data and finetuning them on domain specific dataset for Q/A task.

II. Data

Dataset used for this project is emrQA: A Large Corpus for Question Answering on Electronic Medical Records. Electronic medical records consist of important information about patient's health in the form longitude clinical narratives that is unstructured and collected over time. Physicians often find it important to query information about patient from EMRs for e.g. if the physicians want to know if the patient has any allergies? The answer is available not just as a single clinical note but as a entire timeline of the patients health. In order to take appropriate treatment, it is important to know all the allergies experience by the patient over the years. But in vast majority of case this information is not readily available to physicians which they are looking in EMR. To address this problem there is a need to build a QA system and to train such a system there is a need of such QA dataset. To address this problem emrQA a question answering was developed by students at UIUC and researchers at IBM. This dataset is a closed dataset and is available only available through i2b2 challenges for research purpose.

Some statistics of the current version of the generated data:

Datasets	QA pairs	QL pairs	#Clinical Notes
i2b2 relations (concepts, relations, assertions)	1,322,789	1,008,205	425
i2b2 medications	226,128	190,169	261
i2b2 heart disease risk	49,897	35,777	119
i2b2 smoking	4,518	14	502
i2b2 obesity	354,503	336	1,118
emrQA (total)	**1,957,835**	**1,225,369**	**2,425**

We have used relation dataset, which is one of the 5 subsets of emrQA dataset and has majority of question answer pairs and their format is consistent with the span extraction task, which is more challenging and meaningful for clinical decision-making support.

III. Architecture:

In 2018 Researchers at Google released BERT a Bidirectional Encoder representation from Transformers which is a groundbreaking state of the art model. Bert is used for variety of NLP task as It is trained on a corpus of Wikipedia text and book corpus which is massive amount of data. BERT is a language model which is context free and fully bidirectional unlike previous state of the art model which were context based i.e. to predict the next work you need to have information about the previous tokens in the sequence. BERT is a language model that can look text sequence during training from either left-to-right or combined left-to-right and right-to-left. This means we can now have a deeper sense of language context and flow compared to the single-direction language models. Instead of predicting a next word BERT model randomly mask 15% of words randomly in the sequence during training and try to predict them. BERT also incorporates next sentence prediction where it receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. The sentence is broken and are separated by [SEP] token and the beginning of the sentence is assigned a [CLS]token and based on this the BERT tries to predict which sequence comes first.

There are 2 variants of BERT model called BERT base and BERT Large.

Configurations of BERT Models:

1) **BERT-Base, Uncased**: 12-layer, 768-hidden, 12-heads, 110M parameters
2) **BERT-Large, Uncased**: 24-layer, 1024-hidden, 16-heads, 340M parameters
3) **BERT-Base, Cased**: 12-layer, 768-hidden, 12-heads , 110M parameters
4) **BERT-Large, Cased**: 24-layer, 1024-hidden, 16-heads, 340M parameter

BERT ARCHITECTURE:

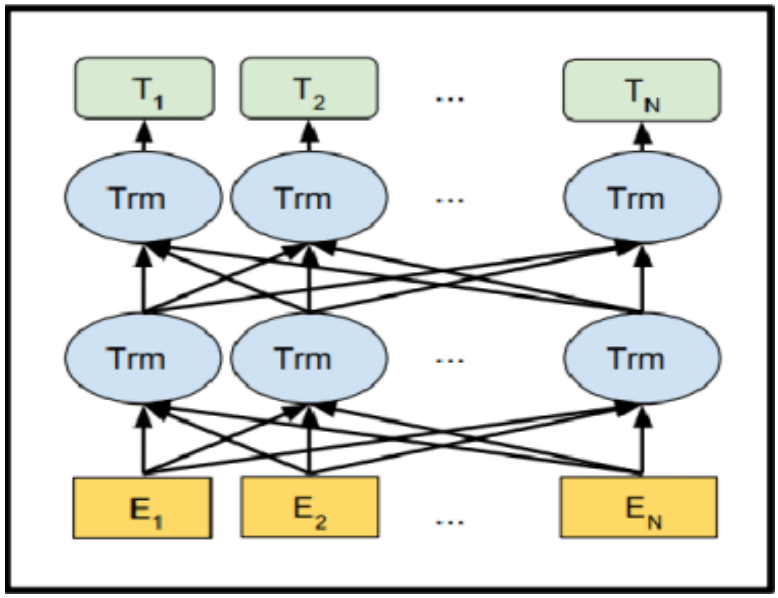

Fig. 1 A visual Representation of BERT Architecture

We have made use of pre-trained BERT base model as training BERT Large is computationally expensive.
The other models used in this comparison are Clinical BERT and Bio BERT: pretrained biomedical language representation model for biomedical text mining. Each model has the same architecture as the base model. The key difference between different model is that Bio BERT is pretrained on PubMed dataset along with Wikipedia and book corpus. Clinical BERT model was trained on all notes from MIMIC III, a database containing electronic health records from ICU patients at the Beth Israel Hospital in Boston, MA.

IV. Implementations:

To be able to reproduce our work, place data.json file in data folder and run the main file. The dataset_prep.py file pre-processes the data, filters a subset of data from emrQA dataset (relation dataset) and breaks them into train, dev and test datasets
The dataset was preprocessed to be of the form of SQUAD dataset to input to the model. The data is then used as input for finetuning on model using run squad.py (official file by google BERT for train/inference) for Q/A task. The Bert model tokenize every input to the model and adds special tokens like CLS and SEP between the start of the sequence and SEP between the two sequence. The sequence starts with the CLS token question token SEP token followed by tokens for context. At the output, the Model will predict the answer with highest scores.
Then using official script provided by SQuAD, we evaluate the results of predictions from test dataset using Exact Match and F1 score metric

Note: Since data is not publicly available, we do not have the permission to share it with the project. Also since pretrained models are very large, they have not been included but can be downloaded from their github pages mentioned in the references

V. Evaluation:

For Evaluating the model performance, we have used the Official Evaluation script (evaluation v1.1) provided for SquAD dataset. The evaluation script gives exact match and F1 scores. The Exact Match metric measures the percentage of predictions that match any one of the ground truths answers exactly. The F1 score metric is a looser metric measures the average overlap between the prediction and ground truth answer. We performed fine tuning on the various pre trained model by using various hyperparameters like batch size, learning rate, training epoch, max length for the sequence. We found the following best parameters i.e batch size=12, learning rate=6e-5, number of training epochs=2.0, maximum length =384 and below are the exact match and F1 scores for our dataset. We started with BERT base model and our hypothesis was that the results may not be good as the pretrained model was not trained on domain specific dataset and it was the least performing model. We than finetune on Bio BERT and Clinical BERT and the results we got were better than that of the BERT model. We got the best results on Clinical Bert as it was trained on MIMIC clinical notes. So, we conclude that we can get better results on domain specific dataset if the pre trained model is already trained on the dataset for relevant domain.

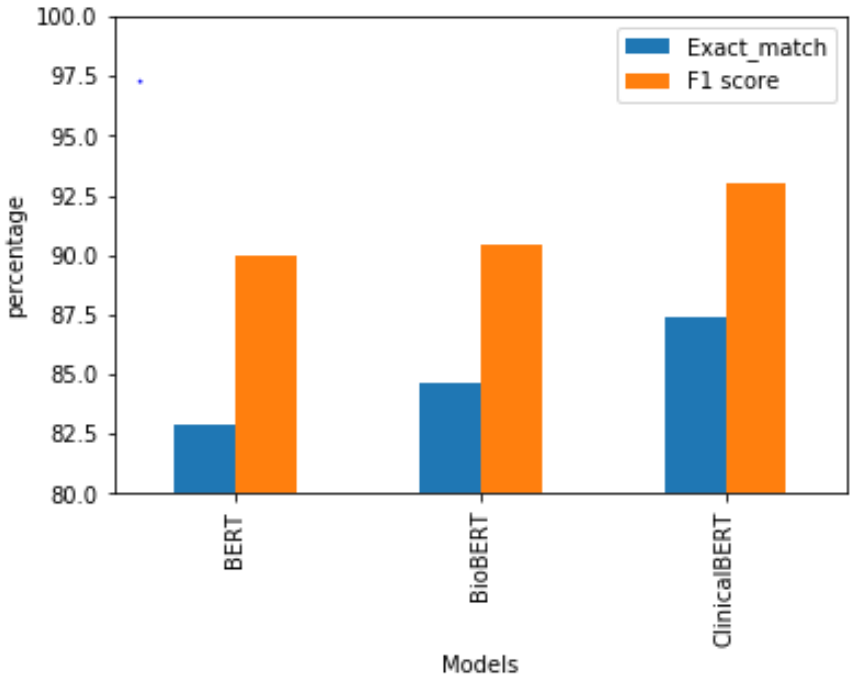

Fig 2. Model Specific Exact_Match and F1_Score

VI. References.

1) emrQA: A Large Corpus for Question Answering on Electronic Medical Records: https://arxiv.org/abs/1809.00732
2) The Stanford Question Answering Dataset SquAD: https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/
3) Bio BERT: a pre-trained biomedical language representation model for biomedical text mining: https://arxiv.org/abs/1901.08746
4) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: https://arxiv.org/abs/1810.04805
5) Publicly Available Clinical BERT Embeddings: https://github.com/EmilyAlsentzer/clinicalBERT
6) https://github.com/allenai/bi-att-flow/tree/master/squad
7) BERT:https://yashuseth.blog/2019/06/12/bert-explained-faqs-understand-bert-working
8) Clinical Reading Comprehension: https://arxiv.org/abs/2005.00574
9) https://medium.com/analytics-vidhya/adapting-bert-question-answering-for-the-medical-domain-2085ada8ceb1
10) https://medium.com/datadriveninvestor/extending-google-bert-as-question-and-answering-model-and-chatbot-e3e7b47b721a
11) https://mccormickml.com/2019/07/22/BERT-fine-tuning
12) https://www.pragnakalp.com/nlp-tutorial-setup-question-answering-system-bert-squad-colab-tpu/
13) https://github.com/xiangyue9607/CliniRC
14) https://hackernoon.com/nlp-tutorial-creating-question-answering-system-using-bert-squad-on-colab-tpu-1utp3352
15) https://towardsdatascience.com/question-answering-with-bert-xlnet-xlm-and-distilbert-using-simple-transformers-4d8785ee762a