Appendices to W266 Final Project (Summer '19)

[Exploring Neural Net Augmentation to BERT for Question Answering on SQUAD 2.0]

Eric Hulburd
ehulburd@berkeley.edu

Suhas Gupta
suhas.gupta@berkeley.edu

Appendix I - Data and code

As mentioned, we used the SQUAD 2.0 dataset available at the SQUAD Explorer homepage. We downloaded BERT base, un-cased model checkpoints from the repository homepage at uncased_L-12_H-768_A-12.zip.

All of our model checkpoints and evaluation results (from will be available for download until August 18, 2019 through out Google Cloud storage bucket.

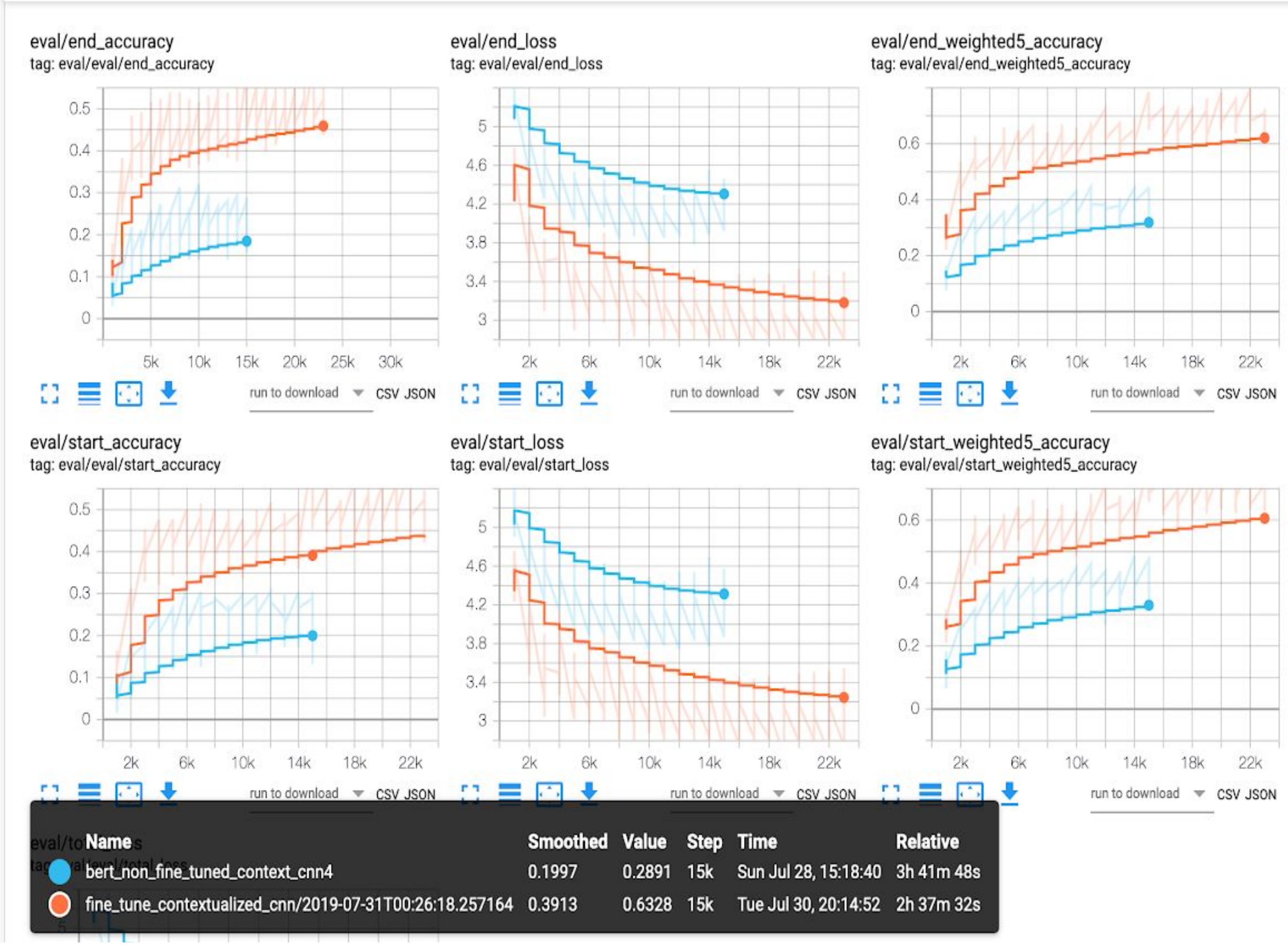
- console.cloud.google.com/storage/browser/w266-final-project-us-central/suhas_gupta/trained_models/?project=w266-240206
  - Each of these named directories contains a config.json file that has hyperparameters for each run.
- console.cloud.google.com/storage/browser/w266-final-project-us-central/suhas_gupta/predictions/?project=w266-240206
  - These include JSON files from our predict.py script, as well as the results of running the evaluate-v2.0.py script from the SQUAD 2.0 download, both with null_score_diff_threshold=0.0 for an initial run, and later using the recommended best null score threshold.

Our code is available at https://github.com/erichulburd/gupta-hulburd-w266-final/. Of note:

- **process_squad.py** - Parses the raw SQUAD JSON files into Example and Feature instances, very similarly to the BERT run_squad.py. This file accepts a fine_tune parameter. When false, the script will save a tf_record of BERT sequence *output* for each example. When true, the script will save a tf_record of BERT sequence *input* for each example.
- **models/** - This directory contains our custom architectures that are applied to the BERT sequence output during either fine tuning or extracted feature training and prediction. They return a tensor of logits for both the start and end indices for each example.
- **train.py** - Reads the resulting tf_record files resulting from the above script, reads a provided config file and initializes a model with the appropriate architecture accordingly. The model checkpoints are saved to a time stamped directory along with the corresponding configuration.
- **predict.py** - Re-initializes by the model checkpoints and config.json file and writes answer predictions for each example resulting from the process_squad.py script. Note, by default, it will attempt to read pre-processed records from the SQUAD 2.0 dev dataset.
- evaluate-v2.0.py - This script was downloaded from the SQUAD 2.0 website and generates an eval.json file with accuracy and F1 scores based on the prediction files generated by predict.py

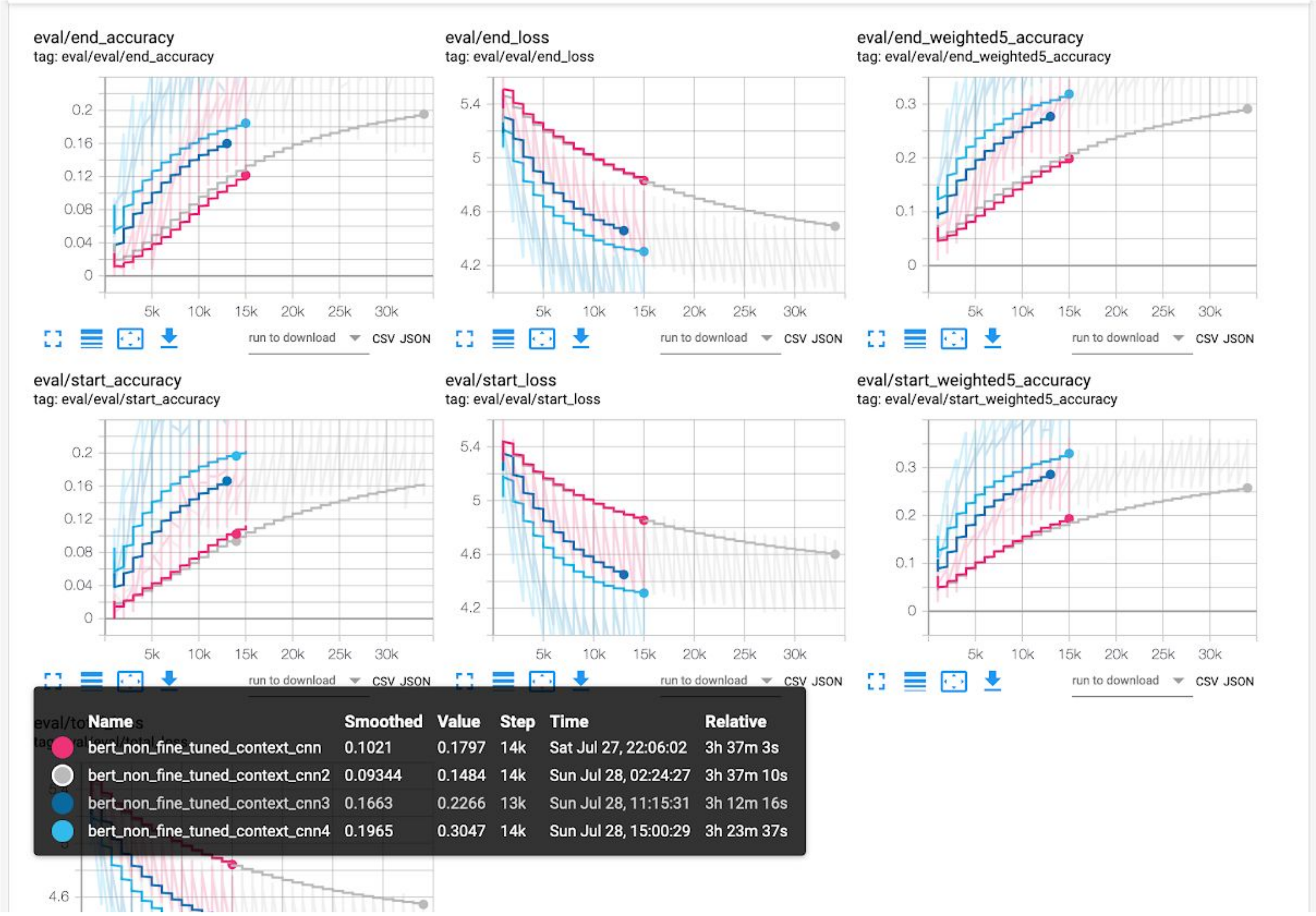
Appendix II - Training and validation run details

Below we include some Tensorboard screenshots of various runs of different architectures and hyperparameters. Note, the screenshots include evaluation metrics run on 10% of the training data we set aside for evaluation. All hyperparameters are for all runs included below are available from the public Google Cloud Storage bucket as described in Appendix I.



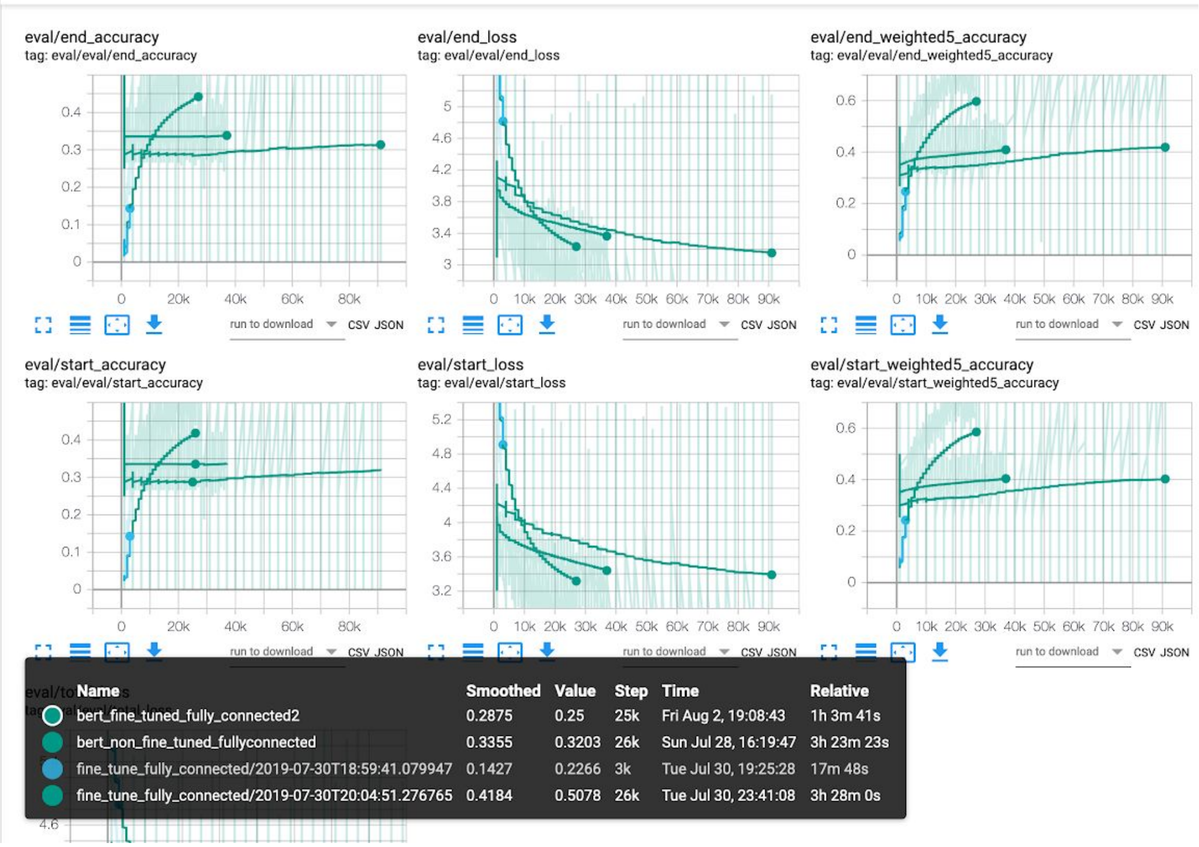
	Name	Smoothed	Value	Step	Time	Relative
●	bert_non_fine_tuned_context_cnn4	0.1997	0.2891	15k	Sun Jul 28, 15:18:40	3h 41m 48s
●	fine_tune_contextualized_cnn/2019-07-31T00:26:18.257164	0.3913	0.6328	15k	Tue Jul 30, 20:14:52	2h 37m 32s

Figure A1.1: Contextualized CNN architecture. Outperformance of the contextualized CNN during fine tuning (*fine_tune_contextualized_cnn*) vs. extract feature run (*bert_non_fine_tuned_context_cnn4*).



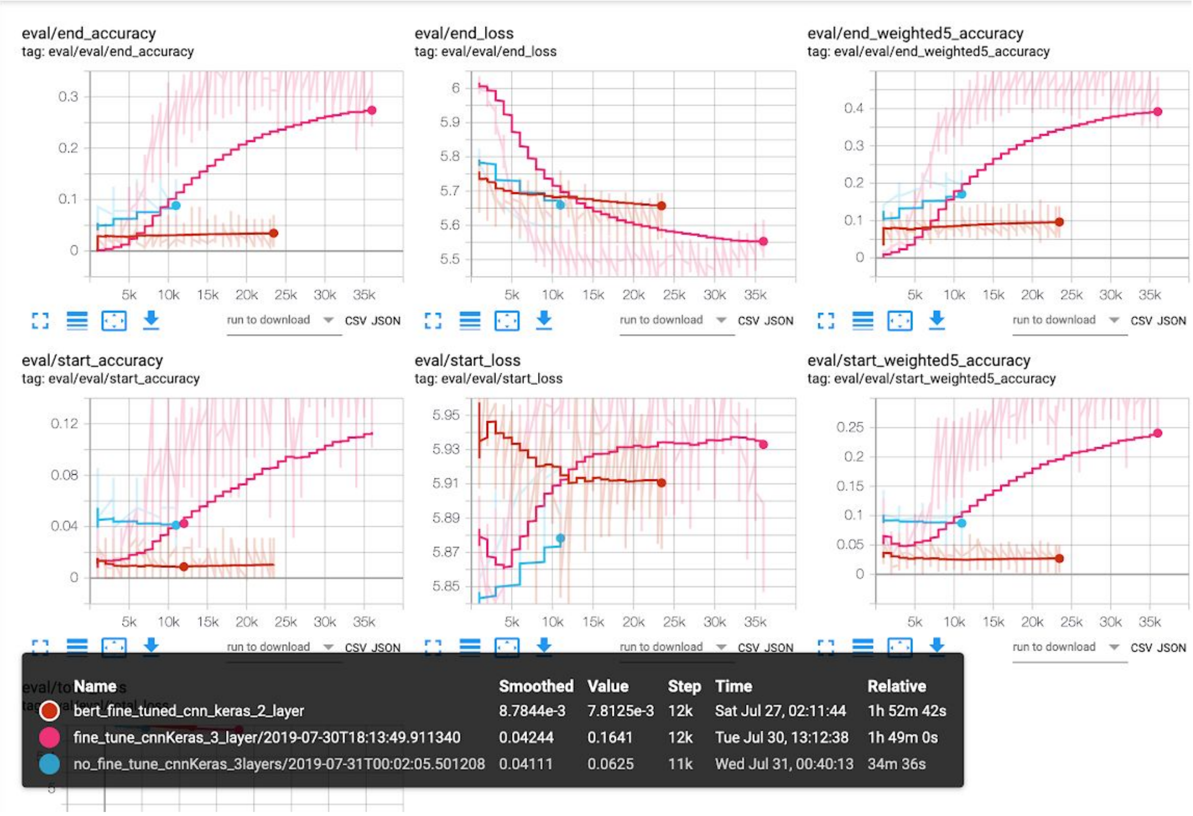
	Name	Smoothed	Value	Step	Time	Relative
●	bert_non_fine_tuned_context_cnn	0.1021	0.1797	14k	Sat Jul 27, 22:06:02	3h 37m 3s
○	bert_non_fine_tuned_context_cnn2	0.09344	0.1484	14k	Sun Jul 28, 02:24:27	3h 37m 10s
●	bert_non_fine_tuned_context_cnn3	0.1663	0.2266	13k	Sun Jul 28, 11:15:31	3h 12m 16s
●	bert_non_fine_tuned_context_cnn4	0.1965	0.3047	14k	Sun Jul 28, 15:00:29	3h 23m 37s

Figure A1.2: Contextualized CNN architecture. *bert_non_fine_tuned_context_cnn* and *bert_non_fine_tuned_context_cnn2* had a downsizing CNN layer before applying contualized CNN filter generators. The other two runs did have CNN downsizing layers. Additionally, *bert_non_fine_tuned_context_cnn4* had four output channels from the contextualized CNN module, whereas *bert_non_fine_tuned_context_cnn2* had only two output channels.


eval/end_accuracy
tag: eval/eval/end_accuracy
0.4
0.3
0.2
0.1
0
0
20k
40k
60k
80k
eval/end_loss
tag: eval/eval/end_loss
5
4.6
4.2
3.8
3.4
3
0
10k
20k
30k
40k
50k
60k
70k
80k
90k
eval/end_weighted5_accuracy
tag: eval/eval/end_weighted5_accuracy
0.6
0.4
0.2
0
eval/start_accuracy
tag: eval/eval/start_accuracy
eval/start_loss
tag: eval/eval/start_loss
5.2
4.8
4.4
4
3.6
3.2
eval/start_weighted5_accuracy
tag: eval/eval/start_weighted5_accuracy
run to download
CSV JSON

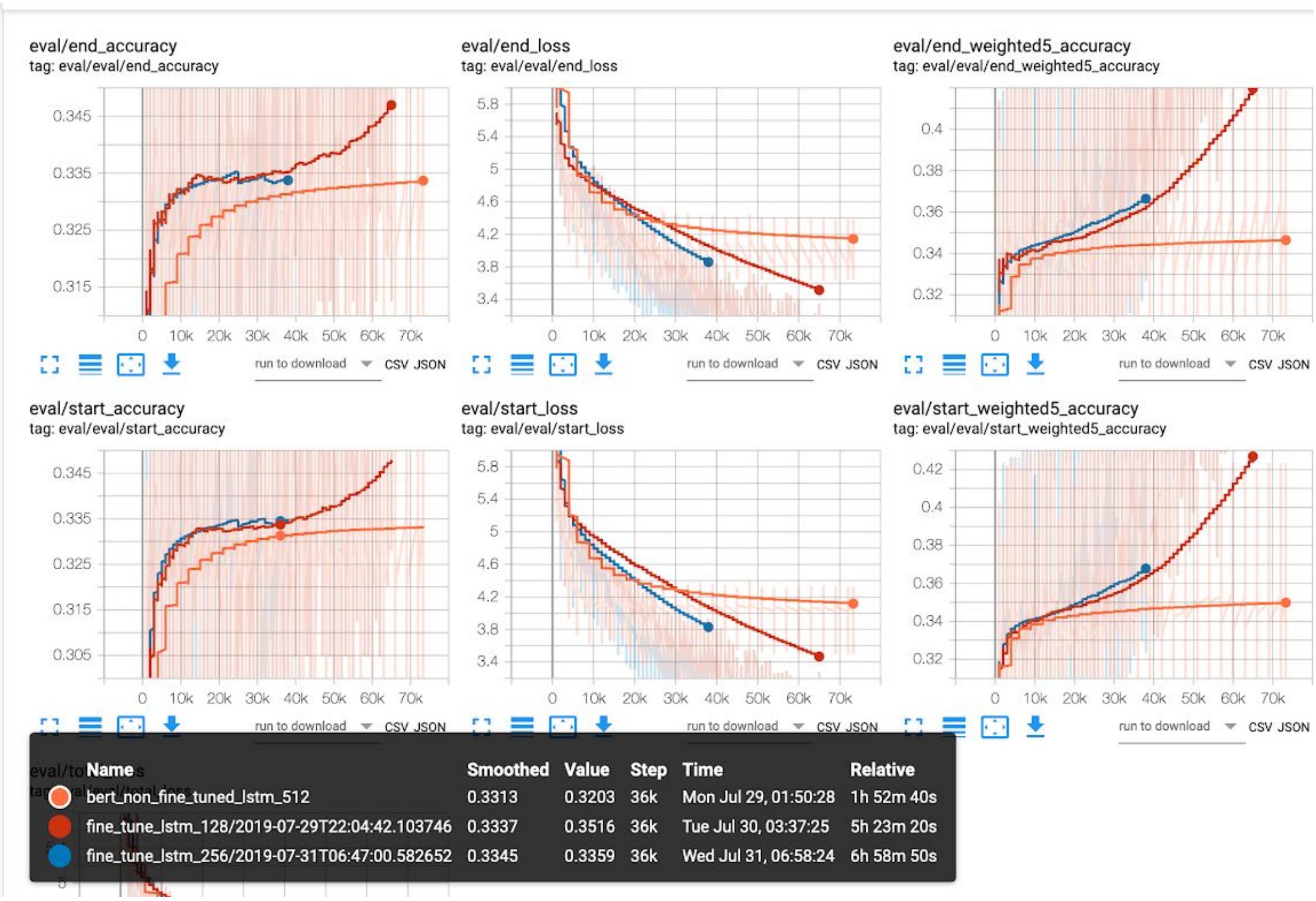
	Name	Smoothed	Value	Step	Time	Relative
○	bert_fine_tuned_fully_connected2	0.2875	0.25	25k	Fri Aug 2, 19:08:43	1h 3m 41s
●	bert_non_fine_tuned_fullyconnected	0.3355	0.3203	26k	Sun Jul 28, 16:19:47	3h 23m 23s
●	fine_tune_fully_connected/2019-07-30T18:59:41.079947	0.1427	0.2266	3k	Tue Jul 30, 19:25:28	17m 48s
●	fine_tune_fully_connected/2019-07-30T20:04:51.276765	0.4184	0.5078	26k	Tue Jul 30, 23:41:08	3h 28m 0s

Figure A1.3: Fully connected architecture. *bert_fine_tuned_fully_connected* multiplies all question tokens by zero vectors to mask them before applying a fully connected layer. *Bert_fine_tuned_fully_connected2* did not apply this question masking and outperformed *bert_fine_tuned_fully_connected*.



	Name	Smoothed	Value	Step	Time	Relative
○	bert_fine_tuned_cnn_keras_2_layer	8.7844e-3	7.8125e-3	12k	Sat Jul 27, 02:11:44	1h 52m 42s
●	fine_tune_cnnKeras_3_layer/2019-07-30T18:13:49.911340	0.04244	0.1641	12k	Tue Jul 30, 13:12:38	1h 49m 0s
●	no_fine_tune_cnnKeras_3layers/2019-07-31T00:02:05.501208	0.04111	0.0625	11k	Wed Jul 31, 00:40:13	34m 36s

Figure A1.4: CNN architecture. We see the fine tuning training outperforming the extracted feature training again. We additionally note that there appeared to be something of a trade-off between start and end accuracy, especially toward the beginning of training. The deeper layer *fine_tune_cnnKeras_3_layer* outperforms the two layer architecture.


eval/end_accuracy
tag: eval/eval/end_accuracy
0.345
0.335
0.325
0.315
0 10k 20k 30k 40k 50k 60k 70k
run to download
CSV JSON
eval/end_loss
tag: eval/eval/end_loss
5.8
5.4
5
4.6
4.2
3.8
3.4
0 10k 20k 30k 40k 50k 60k 70k
run to download
CSV JSON
eval/end_weighted5_accuracy
tag: eval/eval/end_weighted5_accuracy
0.4
0.38
0.36
0.34
0.32
0 10k 20k 30k 40k 50k 60k 70k
run to download
CSV JSON
eval/start_accuracy
tag: eval/eval/start_accuracy
0.345
0.335
0.325
0.315
0.305
0 10k 20k 30k 40k 50k 60k 70k
run to download
CSV JSON
eval/start_loss
tag: eval/eval/start_loss
5.8
5.4
5
4.6
4.2
3.8
3.4
0 10k 20k 30k 40k 50k 60k 70k
run to download
CSV JSON
eval/start_weighted5_accuracy
tag: eval/eval/start_weighted5_accuracy
0.42
0.4
0.38
0.36
0.34
0.32
0 10k 20k 30k 40k 50k 60k 70k
run to download
CSV JSON
5

	Name	Smoothed	Value	Step	Time	Relative
●	bert_non_fine_tuned_lstm_512	0.3313	0.3203	36k	Mon Jul 29, 01:50:28	1h 52m 40s
●	fine_tune_lstm_128/2019-07-29T22:04:42.103746	0.3337	0.3516	36k	Tue Jul 30, 03:37:25	5h 23m 20s
●	fine_tune_lstm_256/2019-07-31T06:47:00.582652	0.3345	0.3359	36k	Wed Jul 31, 06:58:24	6h 58m 50s

Figure A1.5: LSTM architecture. Again we see the superiority of fine tuning relative to extracting features. The 128-hidden unit run appears fairly equivalent to the 256-hidden unit run, however, it runs faster.

Appendix III - Model error analysis

1. Correct predictions:

An instance of correct predictions by our chosen model (contextualized CNN) is shown below. The highlighted text in the paragraph matches the correct answer to the question.

Question: "What fields may pharmacy informatics also work in?"

Context:

"Pharmacy informatics is the combination of pharmacy practice science and applied information science. Pharmacy informaticists work in many practice areas of pharmacy, however, they may also work in **information technology departments or for healthcare information technology vendor companies.** As a practice area and specialist domain, pharmacy informatics is growing quickly to meet the needs of major national and international patient information projects and health system interoperability goals. Pharmacists in this area are trained to participate in medication management system development, deployment and optimization."

Model prediction: “information technology departments or for healthcare information technology vendor companies"

2. Errors with single word answers

Question: "What causes the innate response to be disarmed?"

The type of error with single word answers is when the model find the correct start position but unable to find the answer length correctly. This can potentially be improved with more layers of feature maps that allow for shorter windowing length.

Context:

"Microorganisms or toxins that successfully enter an organism encounter the cells and mechanisms of the innate immune system. The innate response is usually triggered when **microbes** are identified by pattern recognition receptors, which recognize components that are conserved among broad groups of microorganisms, or when damaged, injured or stressed cells send out alarm signals, many of which (but not all) are recognized by the same receptors as those that recognize pathogens. Innate immune defenses are nonspecific, meaning these systems respond to pathogens in a generic way. This system does not confer long-lasting immunity against a pathogen. The innate immune system is the dominant system of host defense in most organisms."

Model prediction : "microbes are identified by pattern recognition receptors"

3. Adversarial no-answer questions

These types of questions are also new in SQUAD 2.0 and are especially difficult for machine translation models to answer since the text is adversarially written to look like the correct answer. More complex neural net architecture may be required to detect these questions correctly.

Question: "What has a stronger association with the MHC:antigen complex than killer T cells?"

This is an unanswerable question on the basis of the text in the paragraph. Our model is unable to identify the impossible question correctly and predicts an answer.

"Helper T cells express T cell receptors (TCR) that recognize antigen bound to Class II MHC molecules. The MHC:antigen complex is also recognized by the helper cell's CD4 co-receptor, which recruits molecules inside the T cell (e.g., Lck) that are responsible for the T cell's activation. Helper T cells have a weaker association with the MHC:antigen complex than observed for killer T cells, meaning many receptors (around 200–300) on the helper T cell must be bound by an MHC:antigen in order to activate the helper cell, while killer T cells can be activated by engagement of a single MHC:antigen molecule. Helper T cell activation also requires longer duration of engagement with an antigen-presenting cell. The activation of a resting helper T cell causes it to release cytokines that influence the activity of many cell types. Cytokine signals produced by helper T cells enhance the microbicidal function of macrophages and the activity of killer T cells. In addition, helper T cell activation causes an upregulation of molecules expressed on the T cell's surface, such as CD40 ligand (also called CD154), which provide extra stimulatory signals typically required to activate antibody-producing B cells."

Model prediction: "Helper T cells"