Self Ensemble Methods with BERT on SQuAD v2.0

Summer 2020: w266
Thaddeus Segura

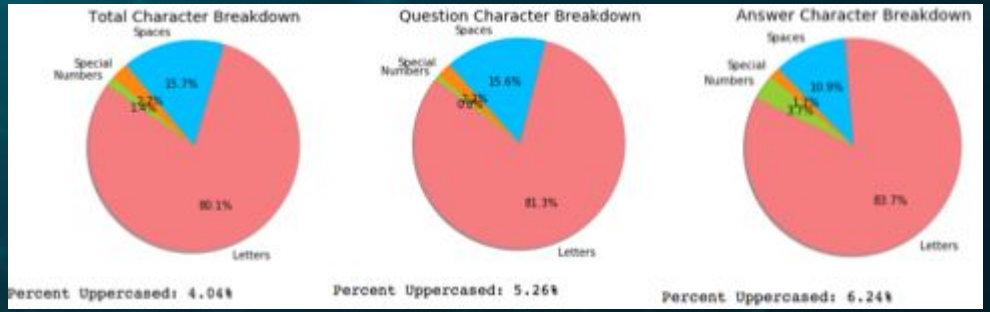
OVERVIEW

SQuAD v2.0

-Scraped from Wikipedia: Context + Question
-Predict the answer by selecting a sequence of text.
-33% of the questions are unanswerable: return only an empty string.

-Training Examples: 130,319
-Dev Examples: 11,873

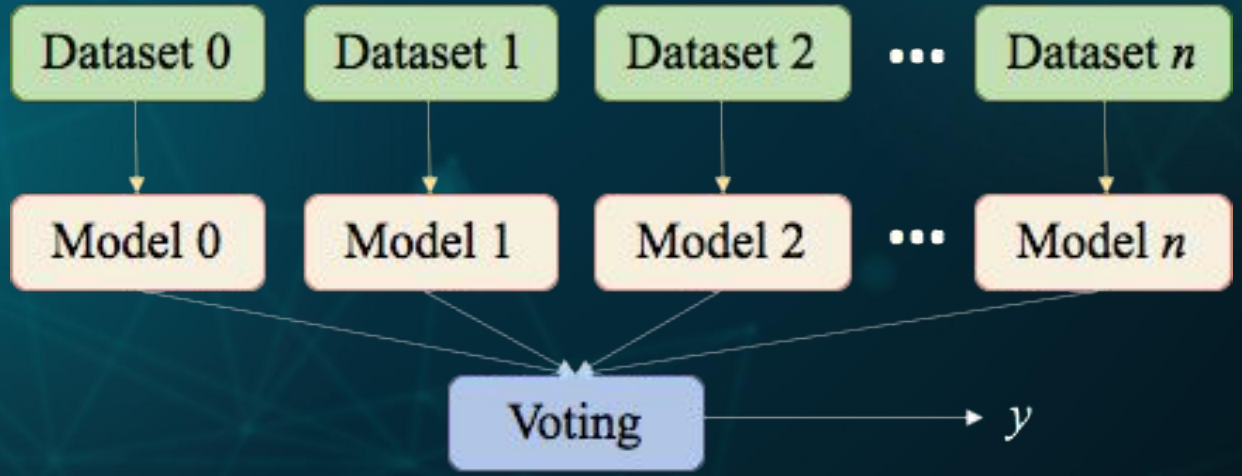
-Average Context Length: 735.5 characters
-Cased words are more likely to appear in question/answer than context. *use BERT Cased.



Motivation

-Initial focus was on Continual Learning.

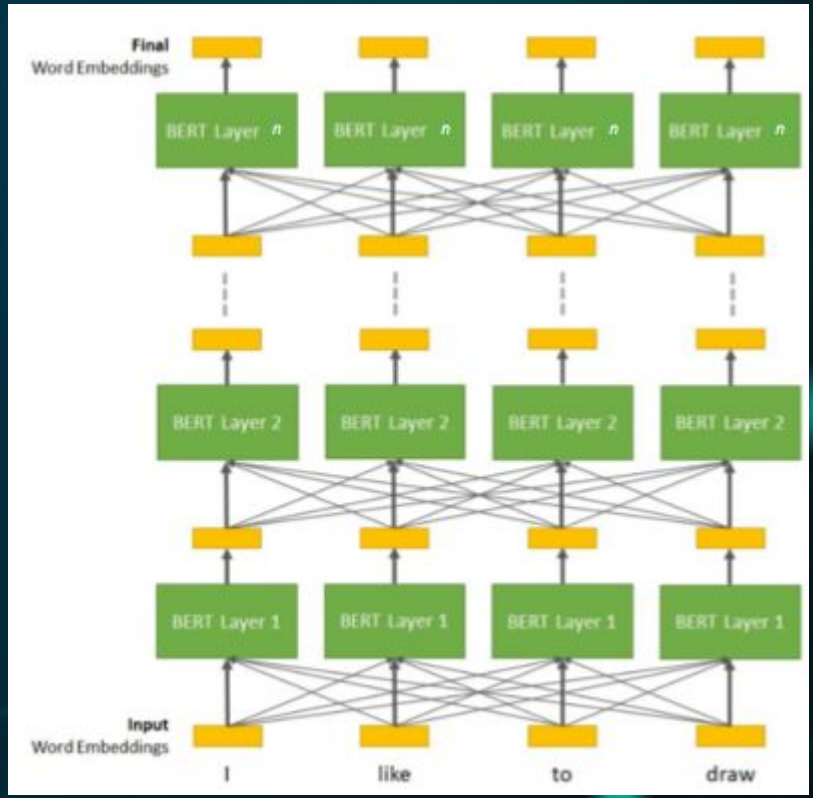
-Could new data be incorporated without requiring retraining a full model by just training a new model on new data, then combine the predictions for an enhanced prediction.



Baseline

-BERT Large: Cased
-24 layers
- ~340 million parameters
-Learning Rate: 3e-5
-Batch Size: 24
-Epochs: 3
-Max sequence length: 384
-Training time on GCP TPU's ~90 minutes

-Total F1: 80.38
-HasAns F1: 83.99
-NoAns F1: 76.77



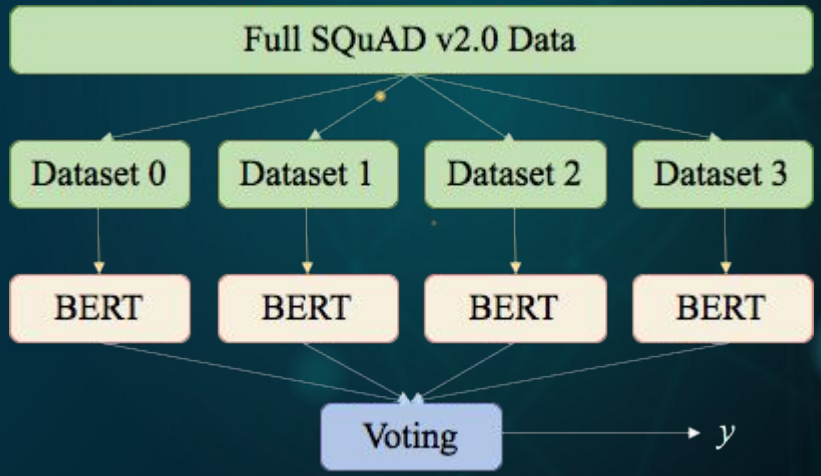
Discrete Data Presentation

-Split the dataset into new discrete datasets.
-Attempted a 4-way and 8-way Split.

-BERT Large: Cased for each dataset.
-Epochs: 4
-Training time on GCP TPU's ~120 minutes

-Total F1: 75.30
-HasAns F1: 82.48
-NoAns F1: 68.14

-More splits = Worse results.



DNN For Model Selection

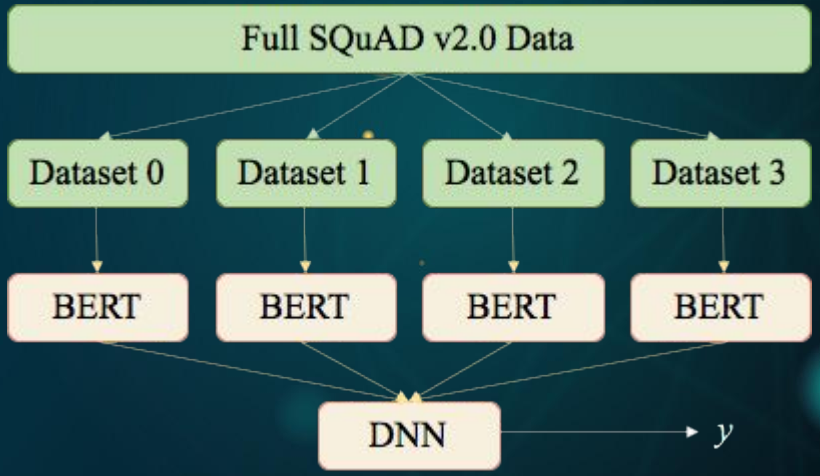
-Use the models generated with the discrete data presentation.

-Generate predictions for all of the training sets questions they had not seen.

-Identify which were correct/incorrect.

-Plug in the word embeddings to a deep neural net to predict which models will get this particular question right/wrong.

-Absolute failure: Deep Neural Network had no understanding of language.



BERT for Multiple Choice

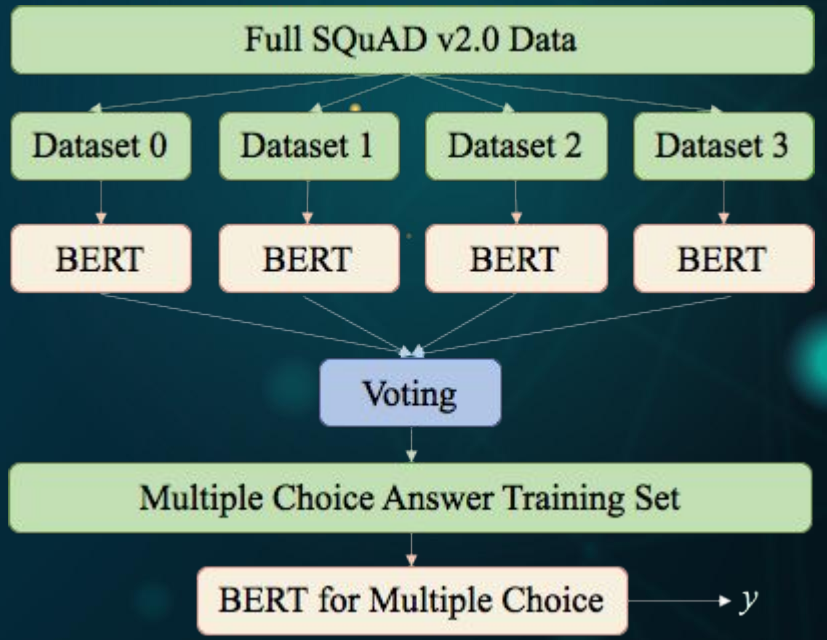
-Replace the Deep Neural Network from the previous approach with another Full BERT model.

-Generate top *n* predictions on each question in the training set to develop a 2nd-Order training set.

-Attempted 2-way and 5-way multichoice.

-Total F1: 67.42
-HasAns F1: 82.48
-NoAns F1: 68.14

*Major Memory issues.
*More choices = Worse Results



BERT for Sequence Classification

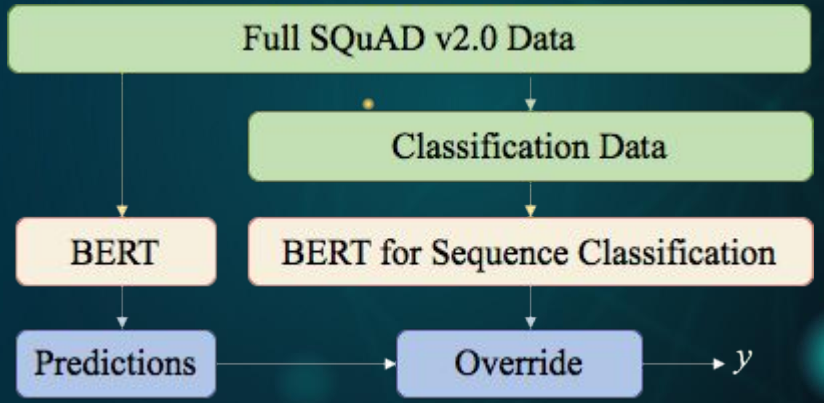
-Classify all questions in the training set as answerable/unanswerable.

-Run Question+Context through the BERTforSequenceClassification model.

-Use those predictions to override predictions made by another BERT model.

-Total F1: 78.37
-HasAns F1: 76.61
-NoAns F1: 80.13 *Highest of all models.

*Still some memory issues.
*Hard to avoid overfitting.



Intelligent Weighting

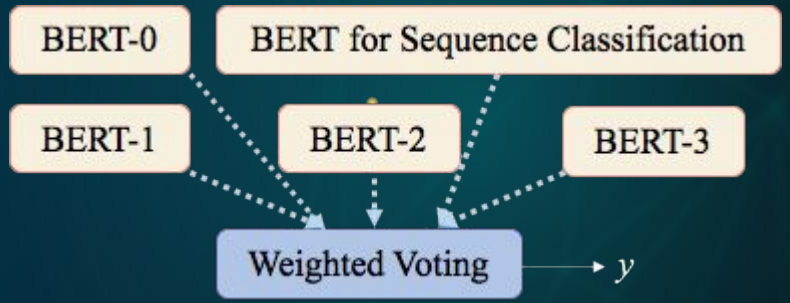
-Factor in Model Confidence and accuracy.

-Manually apply a weight to the classification predictions so that they vote, not override.

-Finally outperformed the baseline.

-Total F1: 81.99 *Highest of all models.
-HasAns F1: 84.10 *Highest of all models.
-NoAns F1: 79.88

*Still major room for improvement by tuning this further.



Results

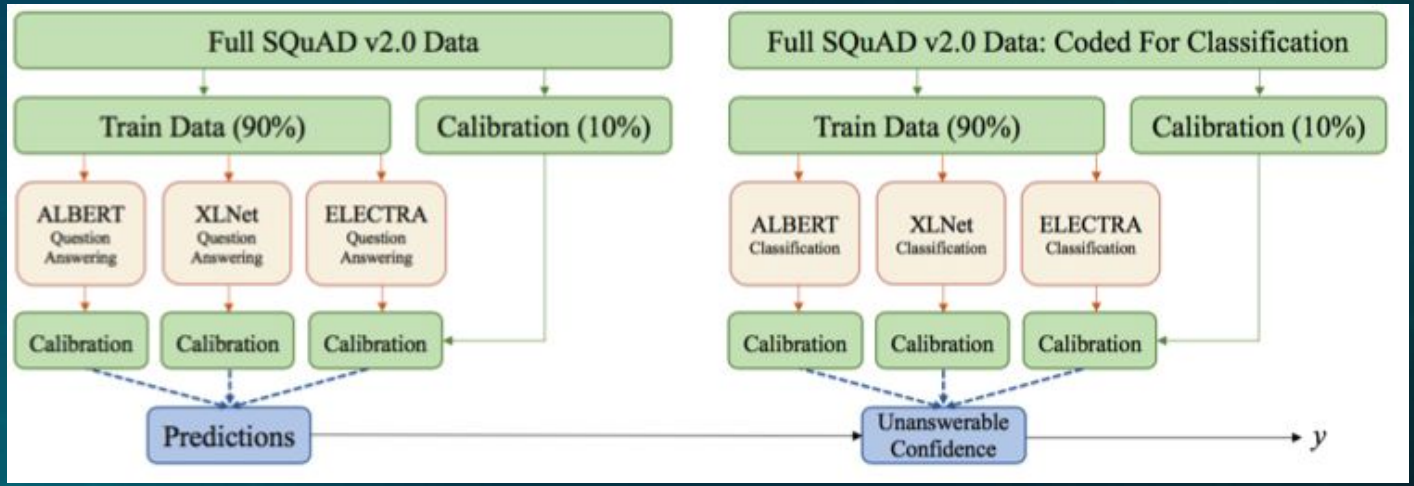
-Intelligent Weighting delivered the best results overall.

-The Classification Model had the best performance on the unanswerable questions.

-Can this be taken further???

Ensemble Type	Total		HasAns		NoAns
	EM	F1	EM	F1	EM/F1
Baseline: BERT Large	77.09	80.39	77.41	83.99	76.77
Simple Voting: 4 Way Discrete Data	71.91	75.30	75.69	82.48	68.14
Multiple Choice: A/B on unsplit data	63.77	67.42	75.67	82.99	51.89
Classification: BERT Base	75.36	78.37	70.58	76.61	**80.13**
Intelligent Weighting: BERT Large + BERT Large unsplit + Classification	**79.04**	**81.99**	**78.19**	**84.10**	79.88

Next Steps



-Proposed Architecture.
-Leverage different models to create the best ensemble predictions.
-Use an ensemble on the classification side as well.
-If I can get the classification accuracy up, I can significantly increase its weight without adverse effects to the other answers.