MODEL COMPRESSION WITH TWO-STAGE MULTI-TEACHER KNOWLEDGE DISTILLATION FOR WEB QUESTION ANSWERING SYSTEM

Paper Link:

https://arxiv.org/abs/1910.08381

Contents:

The Problem Statement

- Deep pre-training and fine-tuning models suffer from **slow inference speed** due to the sheer amount of model parameters
- Applying these complex models to real business scenarios is a challenging but practical problem.
- Model compression methods suffer from **information loss**, leading to inferior models.

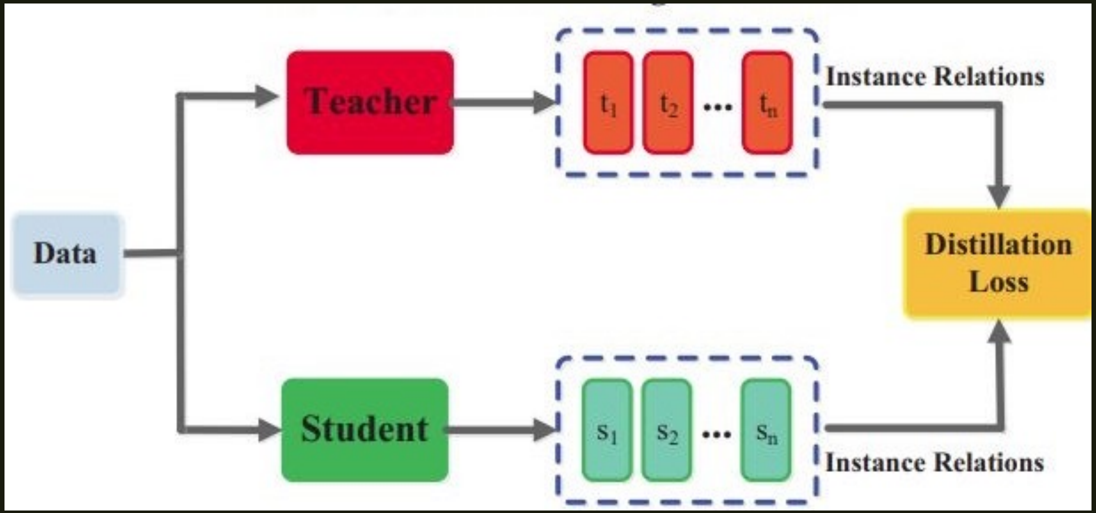
Proposed Solution

Two-stage Multi-teacher Knowledge Distillation method for web Question Answering system

- General Q&A distillation task for student model pre-training
- Further fine-tune pretrained student model with multi-teacher knowledge distillation on downstream tasks (MNLI, SNLI, RTE tasks from GLUE).
- Effectively reduces the overfitting bias in individual teacher models and transfers more general knowledge to the student model
- This method significantly achieves comparable results with the original teacher models, along with substantial speedup of model inference.

Scope of the project

- Decide teacher
- Create student by experimentation
- Implement 1-o-1 model
- Implement m-o-m model
- Implement m-o-1 model
  - Stage 1: 3 BERT Teacher models, 1 student model assuming RTE to be the large corpus
  - Stage 2: Fine-Tune student model on other datasets



KNOWLEDGE DISTILLATION

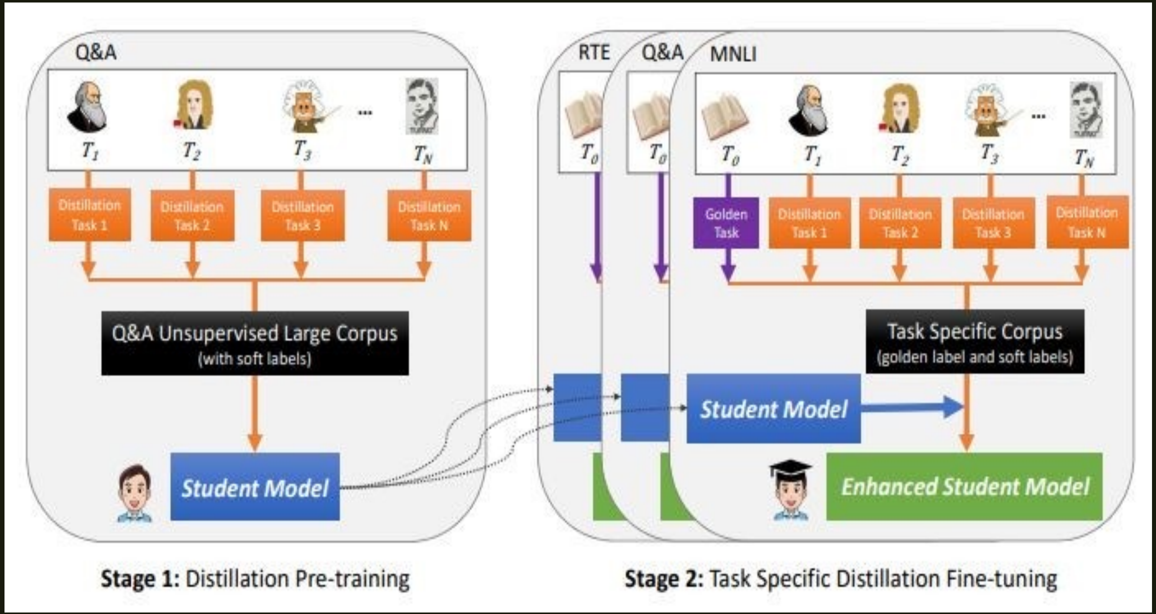
Approaches to Knowledge Distillation

- 1-o-1 model
- m-o-m ensemble model
- m-o-1 model

QUESTION ANSWERING RELEVANCE

Question:	*What can I do when I have headache?*
Passage:	*Drinking warm water mixed with juice squeezed from one-half of a lemon will reduce the intensity of a headache. This particular remedy is beneficial for headaches caused by gas in the stomach. Another option is to apply lemon crusts, pounded into a paste, on your forehead to immediately relieve pain...*
Label:	*Relevant*

THE TWO-STAGE MULTI-TEACHER KNOWLEDGE DISTILLATION APPROACH



Datasets Used

MNLI - Multi-Genre Natural Language Inference

A collection of paired sentences labeled as entailment, contradiction, or neutral

Used for natural language inference tasks.

SNLI - Stanford Natural Language Inference

A collection of paired sentences labeled as entailment, contradiction, or neutral

Used for natural language inference tasks.

QNLI - Question-answering Natural Language Inference

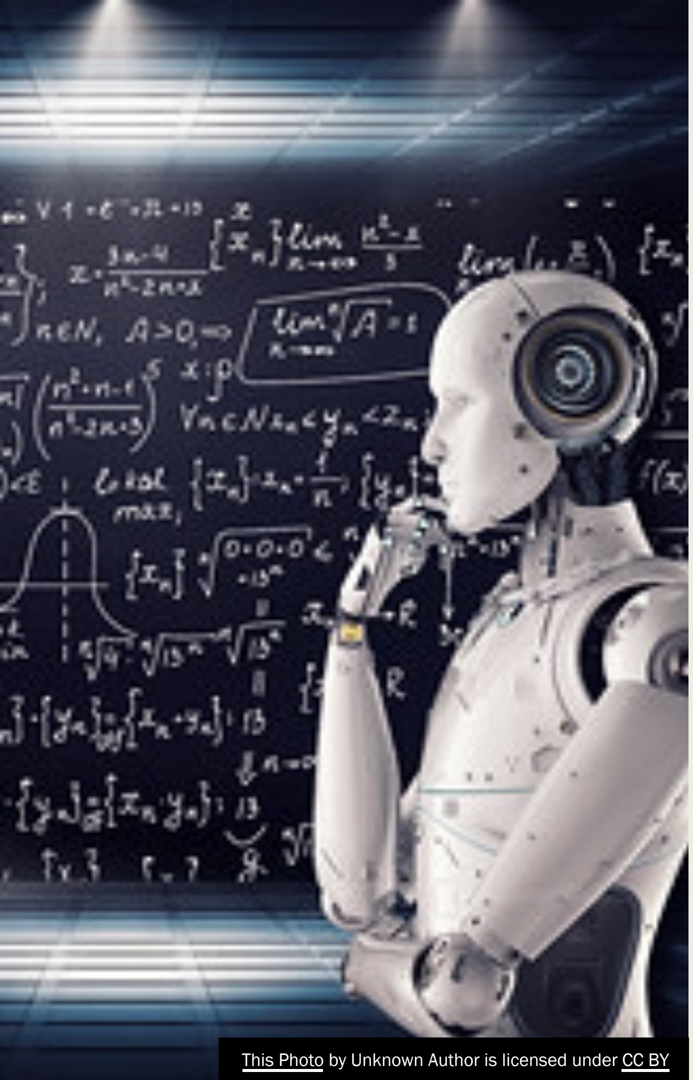
A collection of paired questions and sentences labeled as entailment or not entailment

Used for natural language inference tasks.

RTE - Recognizing Textual Entailment

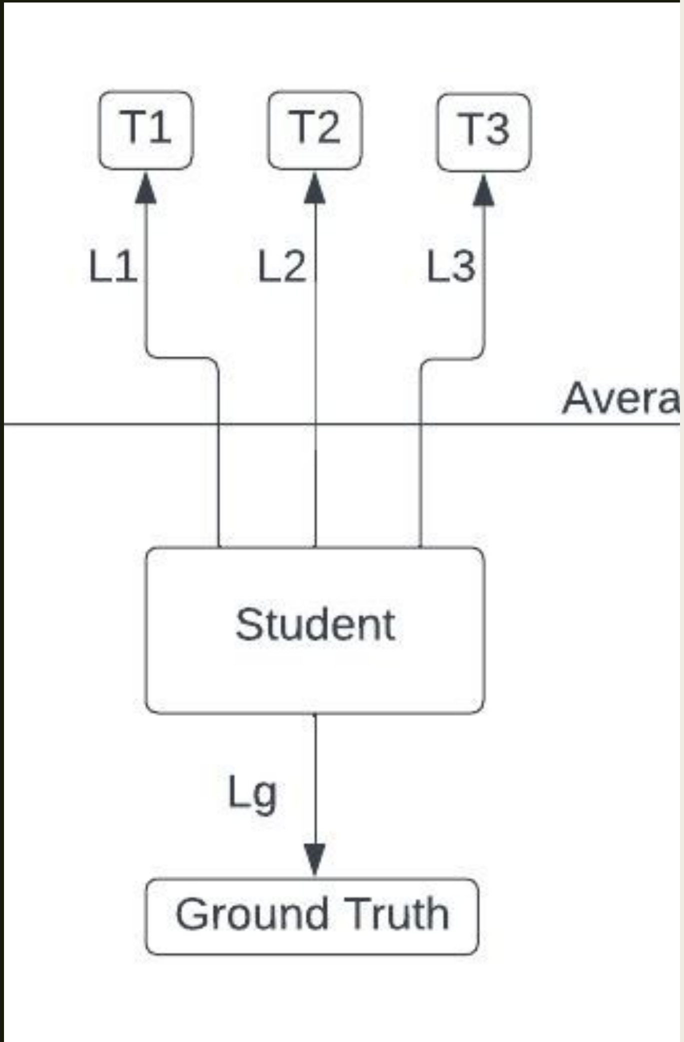
A collection of paired sentences labeled as entailment or not entailment

Used for natural language inference tasks, with a focus on recognizing textual entailment in natural language.



Our implementation

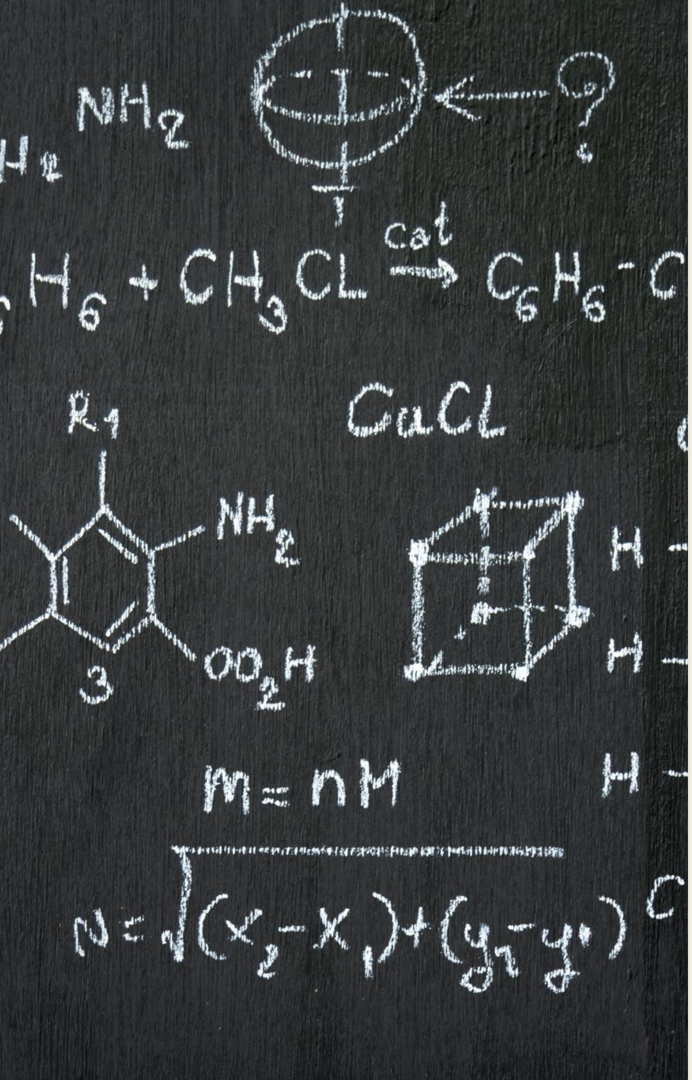
- Prerequisite experimentation *(Results shown later)*
  - Implementation of 1-o-1 KD model on MNLI dataset
  - Implemented m-o-m KD model using ensemble method
  - Trained and tested ensemble model on different datasets
- m-o-1 multi-teacher knowledge distillation to one student assuming RTE as large corpus (*first stage*)
- Fine-tuning the student model on specific task using QNLI (*second stage*)



Concept used

Student loss = $(1 - \alpha)l_g + \alpha * l_s$

Where α is a loss weight ratio



Experiments conducted

- Student model consists of first layer as BERT model layer, and experiments done by increasing the number of subsequent fully connected layers
  - 1 layer
  - 2 layers
  - 3 layers
- Optimizer chosen between ADAM and SGD
- Created 1-o-1 KD model on MNLI, QNLI, SNLI and RTE
- Created m-o-m ensemble model by making multiple teachers and corresponding 1 layer student, and voting logic for final student

Results

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
Adam	0.997	0.36817	0.39545
SGD	0.962	0.2929	0.3163

- Accuracies of experiments on student model (1 bert layer + 1 fc layer) for deciding the optimizer:
  - Teacher model : Bert base uncased
  - Dataset : MNLI

Results

Student Model Architecture	Train Accuracy	Validation Accuracy	Test Accuracy
1 BERT layer + 1 FC layer	0.997	0.36817	0.39545
1 BERT layer + 2 FC layers	0.996	0.37727	0.38183
1 BERT layer + 3 FC layers	0.986	0.36817	0.36817

- Accuracies of experiments on student model (Adam optimizer) for deciding the number of layers:
  - Teacher model : Bert base uncased
  - Dataset : MNLI

Results

- Accuracies of teacher (bert base uncased) - student (Adam optimizer and 1 bert layer + 1 fc layer) model experiments on different datasets:
    - Teacher 1 (T1) : Epochs = 5, Learning rate = 2e-5,
    - Teacher 2 (T2) : Epochs = 5, Learning rate = 3e-5
    - Teacher 3 (T3) : Epochs = 5, Learning rate = 5e-5
    - Student 1 (S1)
    - Student 2 (S2)
    - Student 3 (S3)
    - Student (S) : Majority voting ensemble

RESULTS

Model	MNLI	RTE	SNLI	QNLI
Teacher 1 (T1)	0.5182	0.568	0.54	0.7769
Teacher 2 (T2)	0.6681	0.589	0.5	0.8417
Teacher 3 (T3)	0.5818	0.517	0.6428	0.8417
Student 1 (S1)	0.4275	0.625	0.35	0.5755
Student 2 (S2)	0.3727	0.589	0.438	0.4748
Student 3 (S3)	0.3772	0.611	0.52	0.5755
Student (S) - Ensemble	0.3681	0.5948	0.469	0.5467

RESULTS

Accuracy of m-o-1 model

Creating student model

```
student_11 = MLP(0.1).to(device)
student_11_optim = torch.optim.Adam(student_11.parameters(), lr=1e-4)
train_logits_list_1 = [train_logits_11, train_logits_12, train_logits_13]
val_logits_list_1 = [val_logits_11, val_logits_12, val_logits_13]
test_logits_list_1 = [test_logits_11, test_logits_12, test_logits_13]
train_pred_labels_student_11, val_pred_labels_student_11, test_pred_labels_student_11, train_pred
```

```
Training...........
Checkpoint accessing...........
Resuming training from epoch 21
Epoch 30   Train loss: 0.0290820110142231    Train accuracy: 0.962
Epoch 40   Train loss: 0.02895363214612007    Train accuracy: 0.97
Evaluating on training data..................
1000 1
Training loss: 0.02603114864230156 Train accuracy:  0.969
Evaluating on validation data..................
138 1
Validation loss: 0.04896077340927677 Validation accuracy:  0.5869565217391305
Evaluating on testing data..................
139 1
Testing loss: 0.04710530548644581 Test accuracy:  0.5827338129496403
```

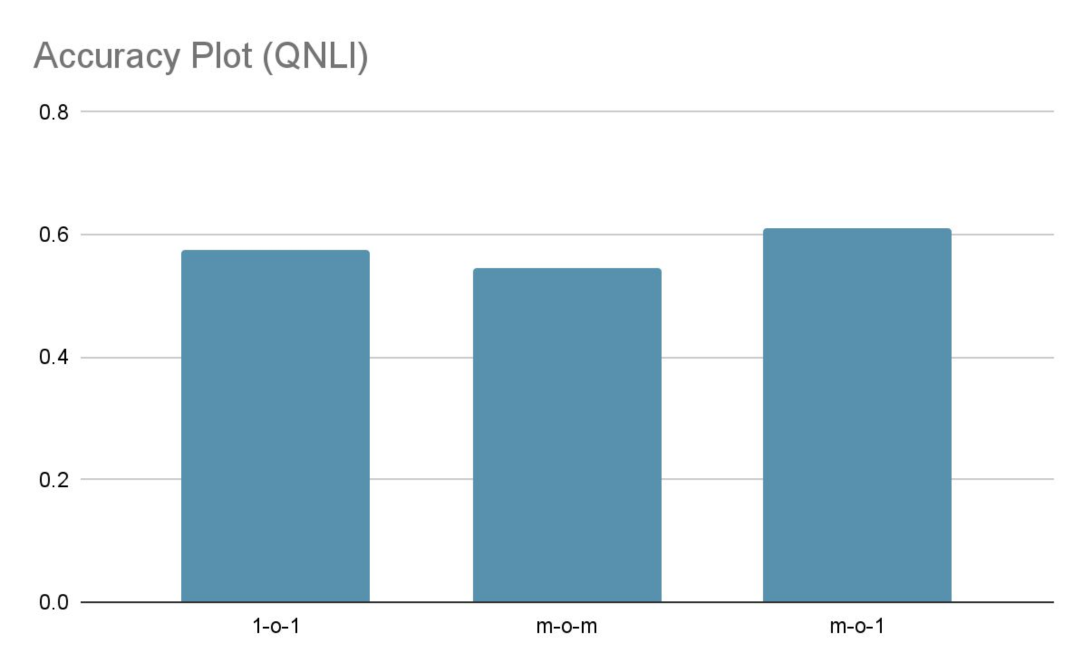
Fine tuning pre-trained student on qnli dataset

```
train_logits_list_2 = [train_logits_21, train_logits_22, train_logits_23]
val_logits_list_2 = [val_logits_21, val_logits_22, val_logits_23]
test_logits_list_2 = [test_logits_21, test_logits_22, test_logits_23]
train_pred_labels_student_12, val_pred_labels_student_12, test_pred_labels_student_12, train_pred
```

```
Training...........
No saved checkpoints to resume
Epoch 0   Train loss: 0.04956310951709747    Train accuracy: 0.509
Epoch 10   Train loss: 0.028006032228469847    Train accuracy: 0.975
Epoch 20   Train loss: 0.027910725146532058    Train accuracy: 0.977
Epoch 30   Train loss: 0.027913129150867463    Train accuracy: 0.978
Epoch 40   Train loss: 0.027914887696504593    Train accuracy: 0.978
Evaluating on training data.................
1000 1
Training loss: 0.025544155269861223 Train accuracy:  0.978
Evaluating on validation data.................
220 1
Validation loss: 0.05101537812839855 Validation accuracy:  0.4818181818181818
Evaluating on testing data.................
220 1
Testing loss: 0.04940701127052307 Test accuracy:  0.5318181818181819
```

RESULTS

Accuracy of m-o-1 model



Comparison

- Comparison between 1-o-1, m-o-m and m-o-1 model

Challenges faced

- In the first stage, a large corpus of Q&A dataset was derived from unlabelled data obtained from commercial search engine using BERT Large. We did not have access to this dataset hence we had to assume RTE as our large corpus (hence the lower accuracies)
- Limited resources with respect to computation power (google colab gpu runtime kept crashing)
- Understanding the loss function of the first stage of multi-teacher knowledge distillation

THANK YOU!