Natural language QA involving Numerical Reasoning

Vamsi Krishna Kanagala

ASU ID: 1218608781

Prathyusha Kodali

ASU ID:1218986912

Chandrahas Chintalaboguda

ASU ID:1218686443

Gayathri Alloju

ASU ID: 1218539582

Sabyasachi Bisoyi

ASU ID:1218272029

Project Definition:

The task of this project is to analyze the existing QA models. Given a passage and a question on the content of the passage, the model should be able to resolve the references in the question and perform discrete reasoning over the content in the given passage to obtain an answer to the question. The questions chosen will enforce the model to understand the semantics of the paragraph, identify the content in the paragraph that is relevant to answering the question by analyzing the structure of the text, extract the arguments and perform discrete numerical operations (which includes addition, subtraction, comparison, sorting or counting etc.) to derive answer.

The first phase of this project requires the synthetic generation of a dataset using template-based QA generation approach. The second phase of this Project aims at exploring the existing Q&A NLP models and testing the given data on these models. It is also verified if the model can handle novel datasets. Accuracy and F-1 scores are chosen as the evaluation metrics to report the results of various experiments.

Synthetic Data Creation:

For this initial phase of the project, it is required to build a synthetic dataset that will be useful to train a QA model. For the implementation of this task, datasets such as AQuA-RAT, DROP, McTaco, math problems in Manhattan and ICSE books and NCERT questions have been explored. Each data sample consists of a question, options and an answer to choose from the options. A template creation approach has been chosen to generate this synthetic data. Using the templates, several numerical questions involving common sense reasoning and mathematical reasoning have been created. Different types of questions have been generated from these templates such as questions based on identification of keywords, questions based on approximation, questions requiring comparison between different values, questions that need performing simple mathematical operations on two or more parameters to obtain a numerical answer etc. Templates are implemented in a way that they can be generalized over multiple domains – ensuring that each template is used to create multiple questions applicable to various domains with different values possible for parameters. The parameter values are read from the excel sheet into data frames and lists and are randomly chosen to create the question. The options are generated for each question in such a way that the correct answer could be found in more than one option. Moreover, unrealistic options and linguistic variations in the options have also been added for certain types of questions. Additionally, the options created are shuffled to reduce any spurious biases and the data samples are copied to the JSON file.

Models experimented:

The main challenge of this phase is to experiment with the existing Q&A models and test the ability of the models to predict the answers on Stage datasets.

Models: The two models used to train the datasets are GenBERT and NumNet.

GenBert: This is a pre-trained LM with generative and extractive abilities. This framework includes injecting numerical skills into a pre-trained LM. In the pre-training phase, two steps are added to pre-train the model over large amounts of synthetic numerical data (ND) and textual data (TD). In the next phase, which is fine-tuning, the model is further fine-tuned over the given Stage datasets.

Pre-training the model with numerical dataset (ND) and Textual Data (TD) provides GENBERT with numerical skills and text synthesizing skills to reach better performance and the ability to generalize to math word problem (MWP) datasets.

Following are the steps executed to train the model and extract the prediction results on given Stage datasets:

1. The given stage train and test datasets are converted to DROP-like format.

2. The numeric and textual datasets used in the pre-training phase are downloaded using bash.

3. Using pip command, the requirements to set up the environment for running models are installed.

4. Features are extracted for both train and test datasets.

5. **Training Step:**

Pre-training: Using the numeric data (ND) and textual data (TD) datasets, GenBERT model is pre-trained. In this step, the weights are randomly initialized, and the updated weights are stored for the next fine-tuning step. The three models generated are:

- GenBERT + ND
- GenBERT + TD
- GenBERT+ ND + TD

Fine-tuning: The above pre-trained models are finetuned using the given stage train datasets. In this phase, the updated weights from the pre-training step are used. Three different models are generated at the end of this phase.

- GenBERT + ND + Stage train
- GenBERT + TD + Stage train
- GenBERT+ ND + TD + Stage train

6. Inference: In this step, the model is tested on the Stage test datasets and the predictions are stored in a JSON file.

7. Evaluation: In this final step, the predictions on the test dataset are evaluated with the gold path(containing actual answers) and the accuracy and f1 score values are obtained.

NumNet: NumNet is a numerical Machine Reading Comprehension (MRC) model. It utilizes a numerically aware graph neural network to consider the incoming input information and performs numerical reasoning over numbers in the question and passage.

Following are the steps used to train the model and extract prediction results on the given stage dataset.

1.All the stage datasets are converted into DROP-like format.

2.The DROP dataset and pretrained RoBERTa model are downloaded using bash.

3.The environments on both agave computing platform and google colab are made ready by installing the requirements.

4.Made changes in RoBERTa config file and vocab files as per requirement.

5.Training Step:

Using DROP dataset: Our model is trained on the DROP dataset and the best model with a high F1 score is saved. We have tried this with one epoch and three epochs.

Using Stage train dataset: In another approach we have trained our model using Stage 1, 2 and 3 dataset and the best model is saved for each stage. We have tried with 10 and 20 epochs. Best F1 score is obtained for 20 epochs.

6.Inference: In this step, the model is tested on the Stage test datasets and the predictions are stored in a JSON file.

7.Evaluation: In this final step, the predictions on the test dataset are evaluated with the gold path and the accuracy and f1 score values are obtained.

Analysis:

This challenge has been broken down into multiple stages. Stage 1, Stage 2, Stage 3 and Stage 4. The complexity of the challenge increases with the stage.

For each stage, we have a train dataset and test dataset. In every stage, a model trained with the corresponding stage dataset is evaluated on a test set that consists of both In-domain and Out-of-domain questions.

For example, a model trained on stage 1 train dataset is used to predict the answers for stage 1 test dataset. This stage 1 test dataset contains both in-domain and out-of-domain questions.

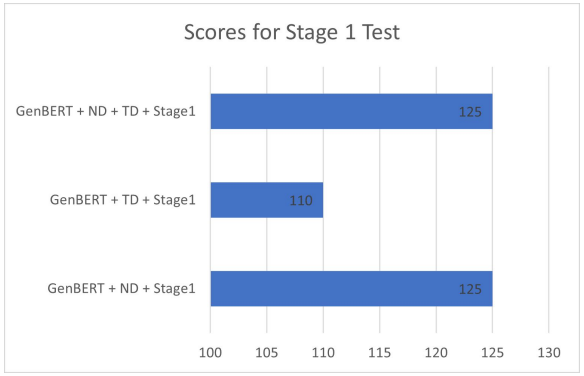
Models are given more credit for correctly solving in-domain and out of domain questions .Final score is computed by aggregating the weighted scores of all the stages.

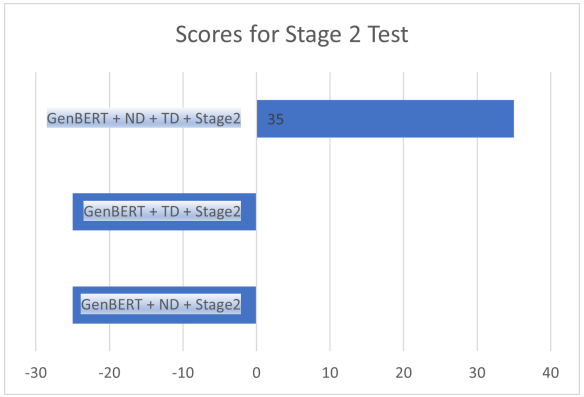
Scoring Criteria for a stage:

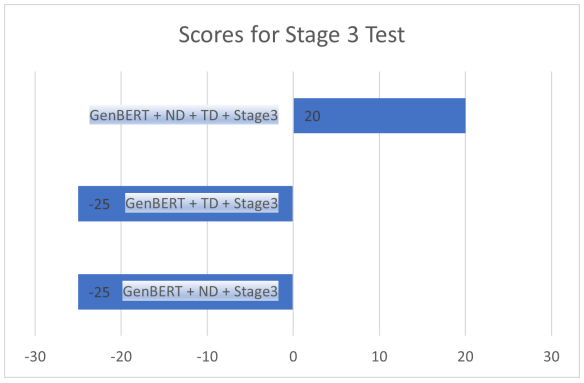
Answer/ Question Category	**In-Domain**	**Out-of-Do main**
Correct	10	15
Incorrect	-5	0

Results for GenBERT Model:

Final Scores: Below are the final scores obtained in each stage for three different models (GenBERT+ ND, GenBERT+ TD and GenBERT + ND + TD). It can be clearly seen the model pre-trained on both ND and TD (GenBERT+ ND+TD) works well in predicting as compared to other models.





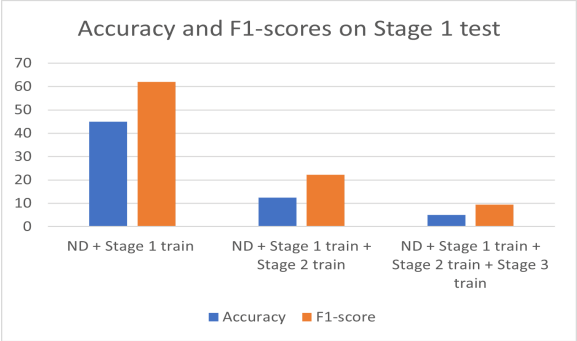


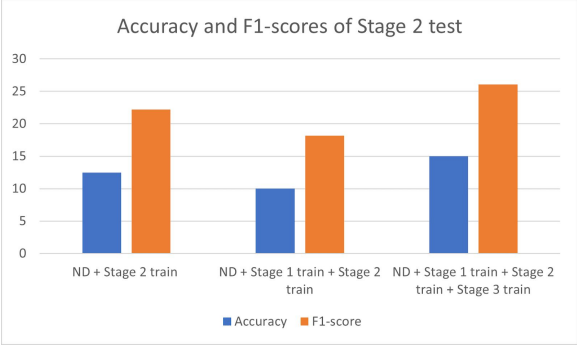
Accuracy and F1-scores: Below are the Accuracy and F1-Score values obtained for different models.

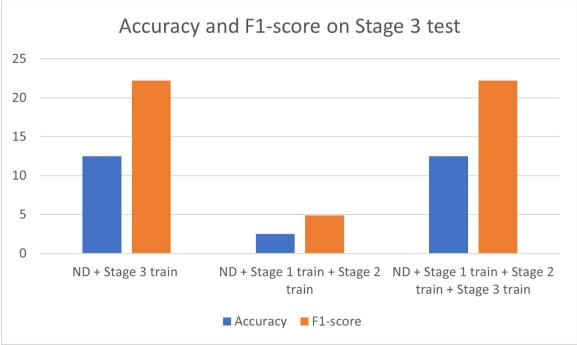
The Accuracy and F1-scores are higher for the models trained with the stage datasets having

in-domain samples rather than those trained with datasets having out of domain samples.

As the number of training examples containing out of domain questions increases, the model performs comparatively poorer as the test error is more.

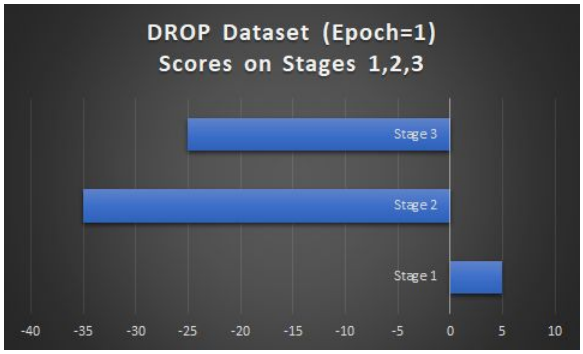


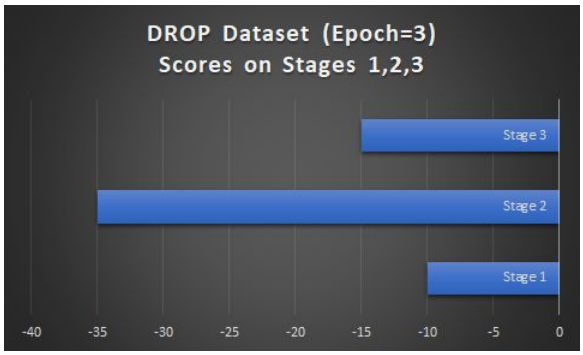




Results for NumNet Model:

Final Scores:





Accuracy and F1-scores: Below are the accuracy and F1 score obtained from NumNet model when trained on DROP dataset, tested with stage test data and next trained on Stage train dataset and tested on stage test data.

We have noticed that though the F1 score is high for DROP dataset training, it is poor in predicting answers for stage test dataset, whereas training on stage dataset yields less F1 score than DROP, but it is good for predicting stage test dataset.

As the complexity of the stage dataset increases it gives poorer performance results.