MID-TERM REPORT

ABSTRACT

This is the mid-term report for the DevRev Problem statement of Inter IIT Tech Meet.

Contents

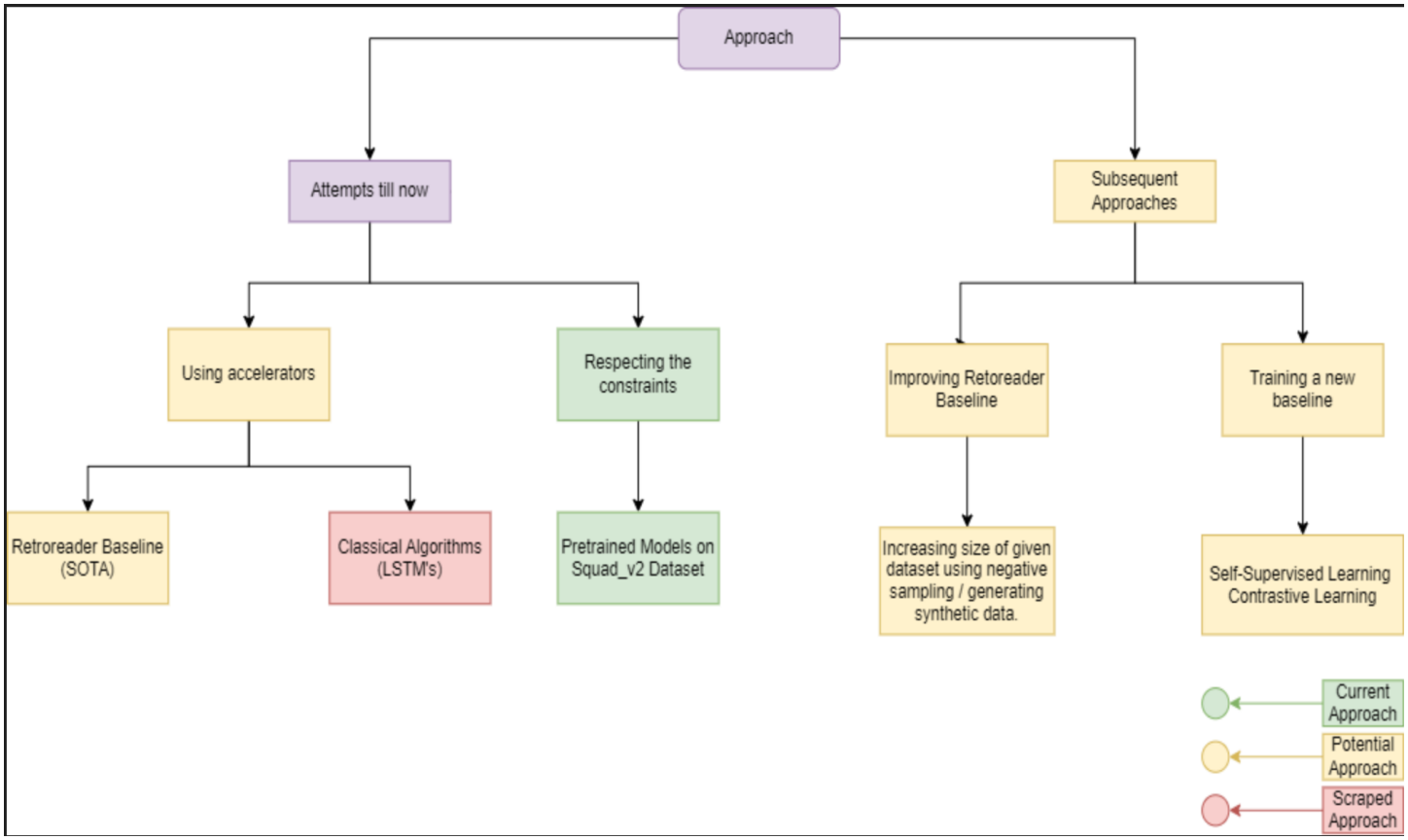
- **Overview** ……
- **Our Approaches** ……..
    - Initial Approach
    - Classical Approach
    - Constrained Approach
    - Data Splits
- **Further Work**
    - Building on Retro Reader
    - Negative Sampling and Synthetic Data Generation
    - Contrastive Learning and Retrieval
    - Self-Supervised Techniques
- **Literature Review**
- **Conclusion**

Overview

Problem Statement: - The problem statement is divided majorly into two parts: (i) Predicting whether the question is answerable from the given set of contexts and (ii) Answering the said question if it is. Question answering is a less researched task in the field of machine learning and natural language processing which makes this problem statement so interesting and open to many approaches.

Constraints: - No hardware accelerators can be used for training or inference. All notebooks have to be run in less than 12 hours on a free colab account.

Basic Outline of Our Approaches:



Our Approaches

Our approach till now has been quite restricted due to the hardware and time constraints. However, we still have managed to try out a few different things. So far, we have divided our attempts into three parts:

1. Temporarily ignoring hardware constraints-

   We started off with researching about the task at hand, we found that the Stanford Question Answering Dataset (SQuADv2) dataset is widely used as a benchmark for Question-Answering tasks and is quite similar to the given data. We found a research paper that had a high F1 score on SQuADv2, Retrospective Reader for Machine Reading Comprehension (Zhang et al. 2021). We implemented the method and had promising results on the given training dataset with an F1 score of around 0.89 using ELECTRA as the backbone. Retro Reader works well because it imitates the human way of text comprehension, by roughly looking for answers in the given passage and then actually finding out the answer.
   The issue with this method was that it was very hardware intensive requiring 4 hours to train on a colab GPU. Due to the accelerator constraint we had to drop this idea.

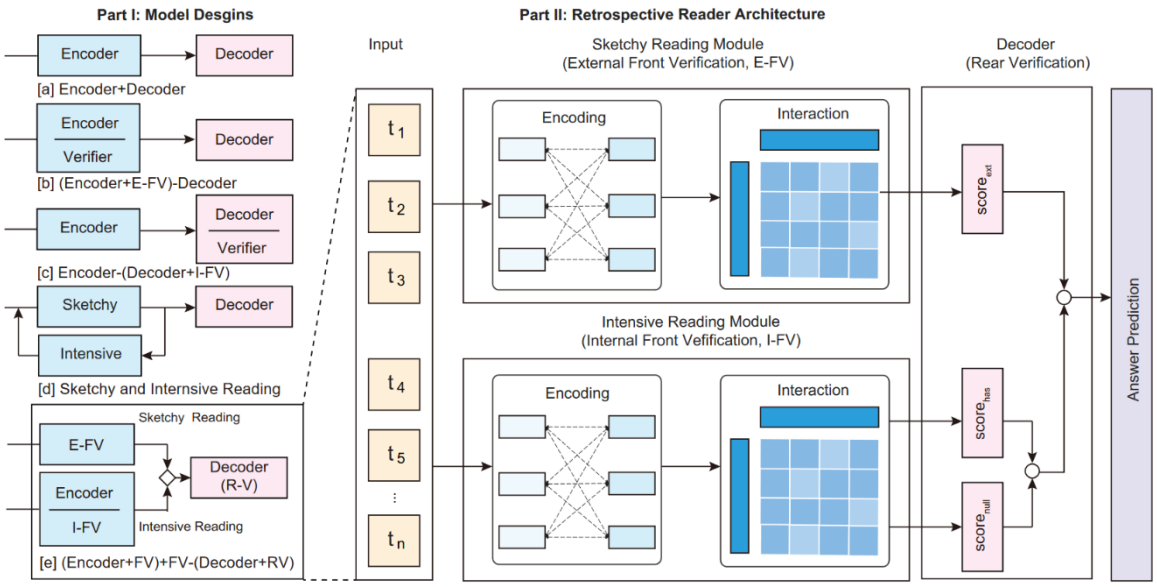


Fig 1. Diagrammatic representation of the Retrospective Reader

2. Classical ML methods with lower computational budget training

   We moved on to using Bi-directional LSTMs taking inspiration from a master's project from Santa Carla University. The method involved training two bidirectional LSTMs for extraction of embeddings and then computing a cosine similarity between them to ascertain whether the question is answerable or not. We tried training this model using only the given constraints i.e. CPU but training took over 30 hours. We then proceeded to use GPU to train it to completion but the F1 score was far too low to be considered. Thus we had to give this method up as well.

3. Using Pretrained Models on SQuADv2 dataset

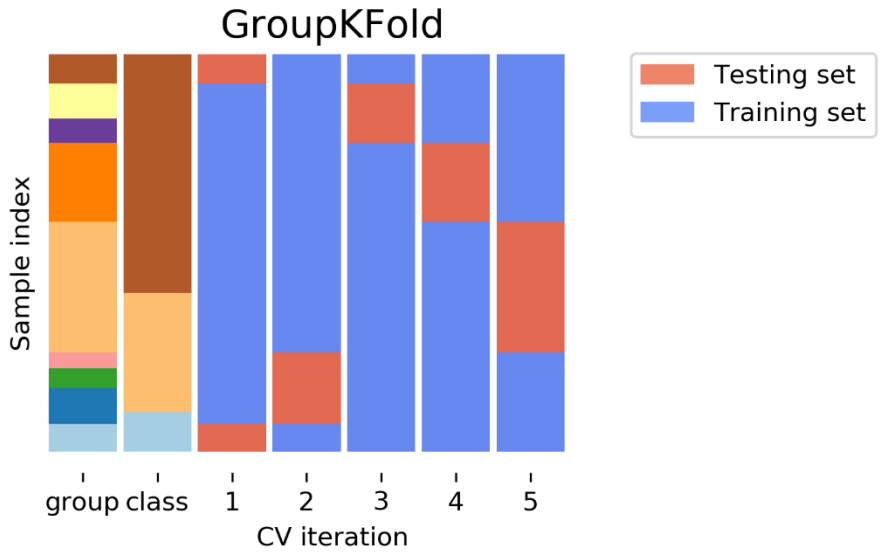
Our current method is to use a **model pretrained on the SQuADv2 dataset**. This dataset is very similar to the given training set, in that it has questions which have to be answered given a context. Considering that we have **significantly lesser data** than in SQuADv2 and the **hardware constraints**, we decided not to fine-tune the pretrained model on our dataset. On evaluating 8 backbones trained on SQuADv2 **using the metric that we will be judged on** we decided to use tiny- RoBERTa as the final model to be used. We found that larger models had a **much larger inference time** and although had better F1 scores their average inference times offset their final metric score so that they would perform worse. This is why we only tested light weight models for inference of the test set. We created the test set using **Group K-Fold** method which is further explained in the next section. Our experimental results using the 8 backbones were as follows: -

Model Name	para score w/o time	qa score w/o time	para score	qa score
MiniLM	0.51	0.51	0.445	0.449
Electra Small	0.56	0.55	0.354	0.355
Electra Base	0.58	0.59	0.373	0.36
ALBERT Base	0.53	0.52	0.31	0.31
RoBERTa Base	0.55	0.54	0.33	0.32
RoBERTa Tiny	0.6	0.59	0.54	0.53
Deberta Small	0.47	0.48	0.24	0.24
Tiny BERT	0.57	0.57	0.52	0.53

As we can see RoBERTa tiny model has a very good para and qa score because of its quick inference time. RoBERTa was originally trained for being more robust than BERT and the tiny model refers to a smaller version of the original model.

Data Split Generation: -

Initially when we planned to train our own model, we split the dataset provided to us to perform validation. As one of the evaluation tasks (Task 1) was going to test our model to generalise to unseen themes we figured a good way to set up data splits would be using Group K-Fold. In this method, the data is divided in such a way that always a certain class of the target variable is unavailable in the training set. This is done to validate the generalisability of the model. When we realised we couldn't train our own model, we reused the splits to evaluate the pre-trained models to get the best one.



Further Work

As far as we can see, there is still a lot of work that can be done on this problem statement provided some relaxation on the hardware constraints is provided. If provided, we plan to set up robust training pipelines using Retrospective Reader as a backbone and build on that. We majorly plan to implement four novel techniques to improve on the current method.

1. **Improving RetroReader Baseline**

   Our major focus will be on using the SOTA method and improving it to fit our task and dataset better. We plan to synthetically increase the size of the dataset to provide more data for training and also reduce data imbalace. The exact method will be elaborated in the next point. Other than increasing training data we will also experiment with multiple backbones for RetroReader, it was originally implemented on the ELECTRA large model but as we have seen large models have a significantly long inference time which makes our evaluation metric take huge hit. So we plan to use light weight transformers to improve the baseline as well.

2. **Negative Sampling and Synthetic Data Generation**

   On doing a cursory EDA, we found out that there is a significant data imbalance in the number of answerable to unanswerable questions (nearly 2:1). Training on this data could lead the model to be more prone to giving answers than refraining. To avoid this we have thought of a way to mitigate this imbalance by creating new data points by matching answerable questions to contexts other than their own ones to create a new un-answerable data point. This is assuming two contextual paragraphs cannot answer the same question. By doing this we can eliminate the data imbalance. This along with other techniques to increase data size we can be sure that our model gets trained to be robust.

3. **Contrastive Learning And Retrieval**

   Contrastive learning will be a big part of the first task of the problem statement. It requires us to retrieve the passage which contains the answer to the given question. We can solve this by using contrastive learning to force an encoder model to learn to represent paragraphs and questions pairs close to each in embeddings whereas far from other paragraph pairs. This will make it easy for the model to retrieve the paragraph with the answer by applying a cosine similarity and finding the closest data point.

4. **Self-Supervised Techniques**

   SSL techniques like MLM and NSP have been extremely successful in training large language models such as BERT to understand language better and become good at encoding textual information. Similar techniques have been developed for aiding the Question-Answering task as well. These include frameworks like elBERto which force the model to fully exploit the additional training signals from contexts containing rich commonsense. We also plan on exploring zero-shot learning for dealing with unseen contexts and questions.

Literature Review

We did a lot of literature review in preparation for this problem statement. The list of all the papers we referred to is given below: -

- Know What You Don't Know: Unanswerable Questions for SQuAD (Rajpurkar & Jian et al. ’18)
- SQuAD: 100,000+ Questions for Machine Comprehension of Text (Rajpurkar & Jian et al. ’16)
- Retrospective Reader for Machine Reading Comprehension (Zhang et al. ’21)
- Long Context Question Answering via Supervised Contrastive Learning (Avi et al. ’22)
- Sentence-aware Contrastive Learning for Open-Domain Passage Retrieval (Wu et al. ’22)
- elBERto: Self-supervised Commonsense Learning for Question Answering (Zhan et al. ’22)
- Self-Supervised Knowledge Triplet Learning for Zero-Shot Question Answering (Banerjee et al. ’20)

Conclusion

For the given task, the hardware restrictions are too stringent to even run inference on a decently performing model much less train it. We have done a lot of work already and plan on doing a lot more provided we get the permission to use the appropriate resources.