

So I want a model which generates HTML code based on prompts, there are some datasets on Huggingface, and [Jawerty](#) which has 2 columns, label(input prompt) and HTML its corresponding HTML code but only 43 rows. Also prominent dataset which can be used is [Alpaca](#) which has 627 rows and 6 columns having instruction and output among them.

Assignment required us to fine tune Falcon 7B/ Llama 7B as they are quite huge models and they will overfit easily. So this task can be categorized into text to text generation. As I need a corresponding html code for given instructions, so it cannot be labeled as question ansrling as it cannot be alone done by context. Also text generation models like Roberta are not suitable as it is not something creative which has to be done and requires to follow protocols for a valid HTML Code. So I have decided to use T5 ( Text-to-Text-Transfer Transformer) developed by Google Research, as it excels in converting input to output tasks.

Since I have to raw train it using Pytorch instead of RFTTrainer( offered by Huggingface), so wre used custom 'MyDataset' class to handle the dataset and Pytorch 'DataLoader' to load the dataset in batches.

I can load the model from transformers library and T5ConditionalGeneration for text generation and T5Tokenizer for tokenization. And since want to generate html code given the label, the mapping betlen them is implicitlity handled during the tokenization, where input texts are labels and target texts are html.

The loss function used is the one provided by T5 model during training, the loss is computed using the 'labels' in forward pass. AdamW is variant of Adam Optimizer algorithm with light decay, which is form of L2 regularization. light decay is used to prevent overfitting by penalizing large lights.

For evaluation I used ROGUE(Recall oriented understanding of gisting evaluation) and not BLEU(Bilingual Evaluation Understudy) , as it focuses on evaluating the overlap of words, phrases or n-grams betlen generated texts and reference texts. This is crucial in content generation. BLEU is focused on precision and is best in cases of machine translation. ROGUE is also language independent and has multiple metrics( ROUGE-N, ROUGE-L, ROUGE-W, etc.) each capturing different aspect. I used ROGUE-L as I wanted to capture the longest subsequence betlen generated and reference text.

I used t5-small as it was lightest model of T5, and I didnt want to it to overfit too much, so I kept the epochs less as well (30). The reason for getting too low of ROGUE is the same, it was trained only on 34 rows and was evaluated on 9 rows(4:1 ratio closely). To incorporate better results we can use Alpaca dataset which has 600+rows.

My choices for dataset and model were made keeping the deadline and GPU limitations,and focused more on the metrics and optimizer and other things mentioned in the assignment.

We can further deploy the model by saving the weights( and store it in weights and biases), or host it on huggingface or the most famous method, use gradio and host it for free for a day.