ICS 661

Advanced AI

Fall 2024

Assignment 3 Report

Grayson Levy

Section 1: Task Description

Part 1:

The task for part 1 was to repeat the task from assignment 2 which was to classify movie reviews as either positive or negative.

Part 2:

The task for part 2 was to fine tune GPT-2 on a dataset containing 1,622 short jokes. Then take the model and generate jokes based on a 3 word input.

Section 2: Model Description

Part 1:

I used the pre-trained LLM distilbert-base-uncased. When loading the model from hugging face I specified "num_labels = 2' this added a output configuration of 2 classes to the model. Which implements cross entropy loss and softmax. This will produce probabilities for either class positive or negative in this case.

Part 2:

I used the pre-trained LLM GPT-2. I added a special token, a pad token, this allowed the model to learn when to stop generating.

Section 3: Experiment Settings

3.1 Dataset Description

Part 1:

The dataset consists of 50,000 movie reviews split equally between test and train sets. Which is then further split equally into 12,500 positive and 12,500 negative reviews in each data set. I further split the training set in a 80/20 split to be used as a validation set while training. The dataset text was cleaned and preprocessed removing special characters, changing to lowercase.

Part 2:

The dataset contains 1622 jokes, the formatting of the jokes was very dirty. I used some NLP techniques to help clean up the text data. I applied the following to the text data, cleaned white space with the punctuations, removed links, special characters, normalized the case, and added a EOS token. The data was not split, all examples were used in the fine tuning of the model.

3.2 Detailed Experimental Setups

Part 1:

I used the smaller Bert model DistilBERT and added a binary classification head to the end of the pretrained model. It was fine tuned over 3 epochs with a learning rate of 1e-4 and a bartch size of 8 with gradient accumulation across 4 batchs which effectively made the batch size 32. I also used 500 warmup steps which is just under 1 epoch. I also implemented a weight decay of .01 to assist with over fitting. After the initial learning rate warmup the Trainer method also implements a linear learning rate scheduler.

Part 2:
I fine tuned the GPT-2 model for 15 epochs with a batch size of 8. The learning rate was set to 1e-4, with a linear learning rate scheduler. I also modified the dropout rate from .1 to .4 for both residual and embedding layers in the model. I added a special token to allow the model to learn when to stop the joke generation. When generating the jokes I set top_k to 50, top_p to .8 and temparature to 1.2 to further help with the quality of the generated jokes.

3.3 Evaluation Metrics

Part 1:
For this part, the model's performance is evaluated using four key metrics which include accuracy, precision, recall, and F1 Score. These metrics provide a comprehensive understanding of the model's classification performance across different aspects.
Accuracy: This metric represents the proportion of correctly classified instances out of the total instances. It is useful for getting a general sense of how well the model performs, but it can be misleading in cases of imbalanced datasets.

Precision: Precision measures the proportion of true positive predictions out of all instances predicted as positive. It focuses on the accuracy of the positive predictions and is particularly important when the cost of false positives is high.

Recall: Also known as sensitivity or true positive rate, recall represents the proportion of true positives out of all actual positive instances. It is crucial in scenarios where missing positive instances (false negatives) is costly.

F1 Score: The F1 Score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall, making it useful when you need to balance false positives and false negatives.

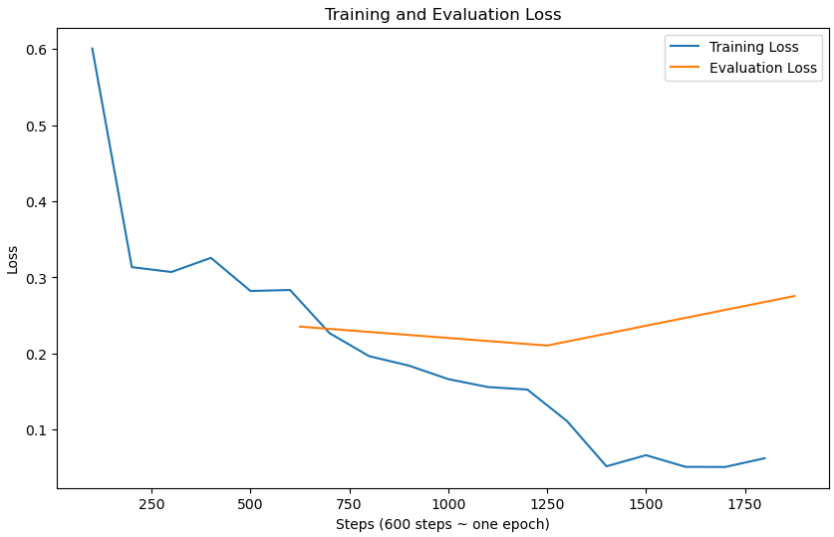
Part 2:
As this is a text generation task, we did not use any evaluation metrics.

3.4 Source Code

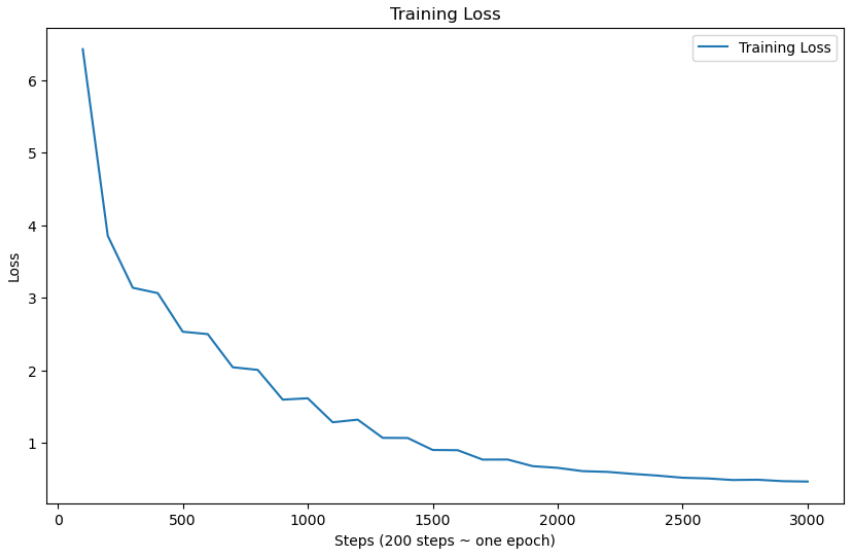
Source code can be found here on GitHub

3.5 Training Convergence Plot

Part 1:



Part 2:



3.6 Model Performance

Part 1:

	Accuracy	Precision	Recall	F1
Training	0.99525	0.9960	0.9943	0.9952
Test	0.9269	0.9231	0.9313	0.9272

Part 2:

Generated Jokes:

Words that exist in the dataset:

Model 1:
Jokes starting with What did the:
What did the mom say when her baby steps outside her house? i think he wants a bath!
What did the french butter say when it got cold? beurre... i can make anything rome
What did the rabbit say to the rabbit hunter? quack, quack, quack.

Jokes starting with What do you:
What do you call a fish with no eyes? a fsh
What do you get when you drop a piano in the middle of the road? a flat minor.
What do you get when you cross kazakhstani with a broad brush? a broad brush

Jokes starting with Why did the:
Why did the rope beech beech tree? because its a bit unwieldy to put up.
Why did the grocery delivery guy get fired? he drove people bananas!
Why did the bullet stay home? because he got a job helping to save the town from destruction.

Model 2:
Jokes starting with What did the:
What did the doctor say to my son about his side of the battle? he just lost his right arm.
What did the french call the sheep that was running around in the mud? a legger!
What did the chicken say to the rabbit? youll never eat ground beef.

Jokes starting with What do you:
What do you call a dog that loves poop? a poop baaan...
What do you call a fish with no eyes? a fsh
What do you call a cow with a hoarse bone? a hoarse beef.

Jokes starting with Why did the:
Why did the chicken coup d'etat fail the house rule? it was ambidextrous.
Why did the boy band break up? because he was a bandit!
Why did the mother of three ask for a divorce?..... because her children were always playing hide and seek.

Words that DO NOT exist in the dataset:

Model 1:
Jokes starting with When i was:
When i was told that by a smart white supremacist, this was the first time i heard that term... well, i muttered.
When i was little, my grandpa wanted to see her married. she wanted to go to a bridal shower and get a divorce. but she was turned away.
When i was little, i didnt know what to watch. my grandma told me this joke the first time around.

Model 2:
Jokes starting with When i was:
When i was little i didnt eat anything but my chromosomes. then i ate a ton of meat.
When i was making a cup of tea i had a huge idiot mistake.
When i was 9, i was diagnosed with gastroenteritis. my body was making excuses for not doing enough gastro. i didnt want to be on the ball.

Discussion on generated jokes:
I tried to achieve two different models. The first model was overfit with a low training loss value and the second model was underfit with a higher training loss. I initial tried for a model that was between the two which is the goal of creating good models, however I was not able to produce such model. Here I will discuss these two models and the difference in jokes they generated.

In model 1, the overfit model, I noticed that many jokes were the exact same as the data which can be expected from a model that is overfit. A few jokes had the same lines but changed minor things about them such as "**What did the rabbit say to the rabbit hunter? quack, quack, quack."** Instead of turkey the model used rabit. The same in this one "**What do you get when you drop a piano in the middle of the road? a flat minor.",** instead of "in a coal mine" it used "in the middle of the road".

In model 2, the underfit model, It still produced jokes with the proper structure but the jokes were a bit gibrish. A couple were still identical but many made no sense. I did notice that the words generated tended to be words used in the training data though.

When observing the generated jokes with words not in the dataset you can see that again the model tried to retain the structure of the jokes but the words generated did not retain the similarity with jokes. They seemed like sentences from a story book mixed with that of a joke structure. This can be seen more with the first model then the second model.

3.7 Ablation Studies

Part 1:

Model trained with stop words removed, batch size of 16, lr 1e-4

	Accuracy	Precision	Recall	F1
Training	0.9928	0.9925	0.9929	0.9927
Test	0.9127	0.9240	0.8994	0.9115

Part 2:

Discussed above