Joke Generator

Timur Aizatvafin, Vsevolod Mikulik, Andrey Starodumov

November 28, 2023

Abstract

This report describes our attempts at joke generation. Briefly speaking, we tried both fine-tuning a pre-trained GPT-2 and training a GPT from scratch (with resources available, of course).

1 Fine-tuning a pre-trained GPT2

As the pre-trained model, we used GPT2LMHeadModel model from an OpenAI adn trained it on the shortjokes dataset.

1.1 Data Preprocessing

Each joke was tokenized by the GPT2Tokenizer with its embedded vocab.

1.2 Training

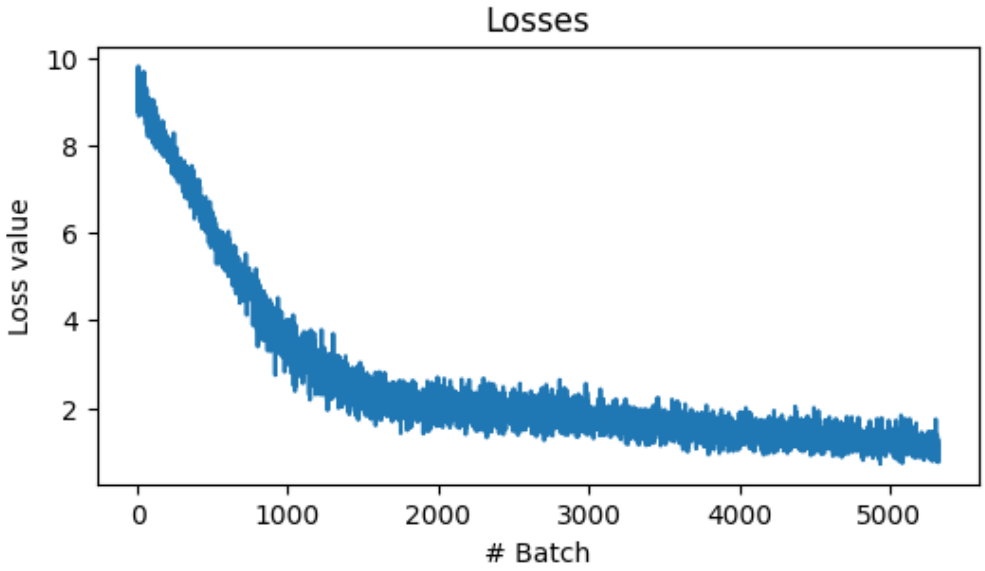
For the training process we used the model function that allows to put the input data and desired output data in the model. After this the model calculates its loss by shifting the desired output data and comparing with model prediction. So the model calculates if it predicted next words correctly.

[Figure 1] shows how loss was decreasing during the training.

1.3 Results

The result is not impressive, but sometimes the model makes funny jokes. Here you can see the few generated jokes:

Figure 1: Losses of pre-trained model



- Input: *"The president kills"*
  Output: *"The president kills the people who make the best decisions"*
- Input: *"What is your"*
  Output: *"What is your greatest weakness? Your inability to read the words on your resume"*
- Input: *"Who is"*
  Output: *"Who is the most popular person in the world? The one who can count"*

2 Training a GPT from Scratch

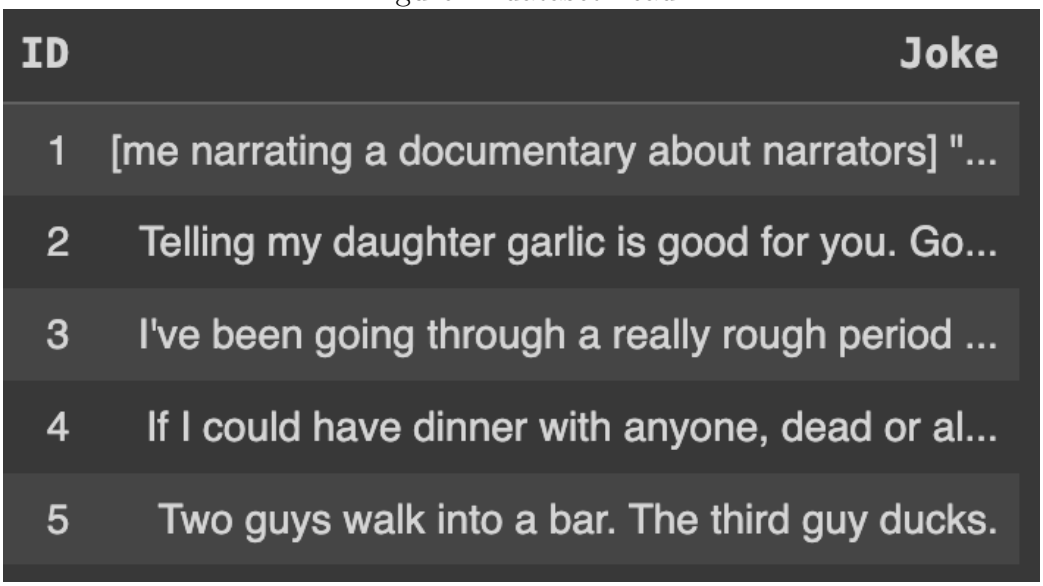
We also came up with an idea of training a GPT from scratch. We found an implementation of GPT and trained it on the dataset of jokes from Reddit.

2.1 Data Preprocessing

Dataset of our choice is a simple .csv file with each entry being a complete joke.

[Figure 2] shows how loss was decreasing during the training.

Figure 2: dataset head



ID	Joke
1	[me narrating a documentary about narrators] "...
2	Telling my daughter garlic is good for you. Go...
3	I've been going through a really rough period ...
4	If I could have dinner with anyone, dead or al...
5	Two guys walk into a bar. The third guy ducks.

In order to train the model, we decided to

1. Tokenize each entry
2. Add "end of text" token to each entry
3. Combine all preprocessed entries into a single long list of tokens

2.2 Training

The model was trained in the following manner:

1. The model receives n tokens as input
2. Token $n + 1$ becomes the label

We've trained the model for 20 epochs on the whole dataset. [Figure 3] shows how loss was decreasing during the training.

Figure 3: Losses of pre-trained model

2.3 Results

Even though most of the jokes are not funny, there are also some that make you laugh simply because of their stupidity

- Input: *"My son"*
  Output: *"My son forgot to put his mind in his pants? He gave him a shit bitch."*

- Input: *"A man"*
  Output: *"A man goes to a doctors... and died of salsa after six boys... It comes back tomorrow."*

3 Conclusion

We tried different approaches to joke generation including training both pre-trained and clean models.

We have also learned how to deploy docker containers with applications that use these models.

It is worth mentioning that quality of joke generation can be improved by

1. Training the model for a longer period of time
2. Using different model architectures
3. Using other datasets in addition to this one

4 References

1. GPT From Scratch
2. Attention is All You Need
3. How to train GPT
4. kaggle gpt2 training notebook
5. GPT2 HuggingFace
6. JokeGeneration