Report

Advaith Malladi

2021114005

Answer 1:

Soft prompt solves the limitations of the discrete hard propmpt because in thr case of the latter, you would have to try a bunch of prompts to find the best prompt you can for the task in hand. Sometimes, the hard prompt you finaliza might not even be the best prompt for the task. What if you could learn the best prompt for the task? In the case of soft prompts, you can just attach a k*768 embedding layer which can be trained just for your task to learn a suitable prompt. Hence it is more flexible.

Answer 2:

Prompt tuning is very efficient compared to fine tuning an entire language model. Compared to training all the millions or billions of parametes in the language model, we just train the k*768 parameters of the prompt embedding layer. Thus this has huge implications for future developments in large-scale language models and their adaptability to specific
tasks.

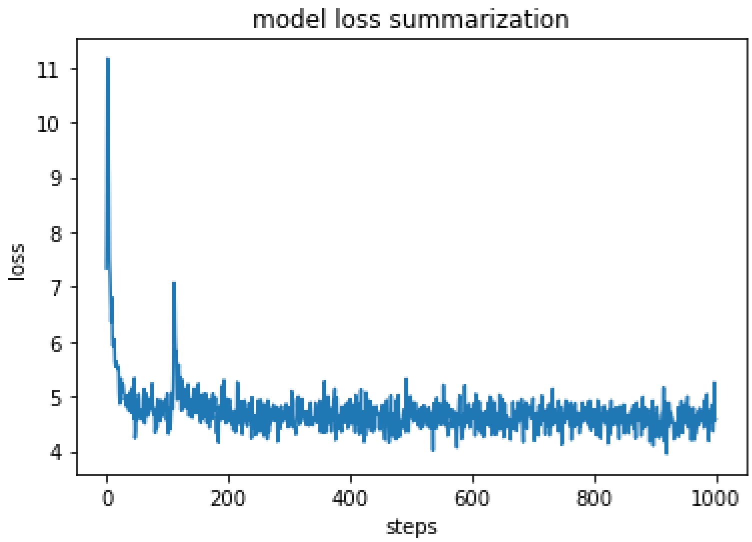
Task 1: summarization

test rouge score: 0.44020787648597154

train loss (avg); 4.526131364822388

vall loss (avg): 4.714538385868073

Graph:



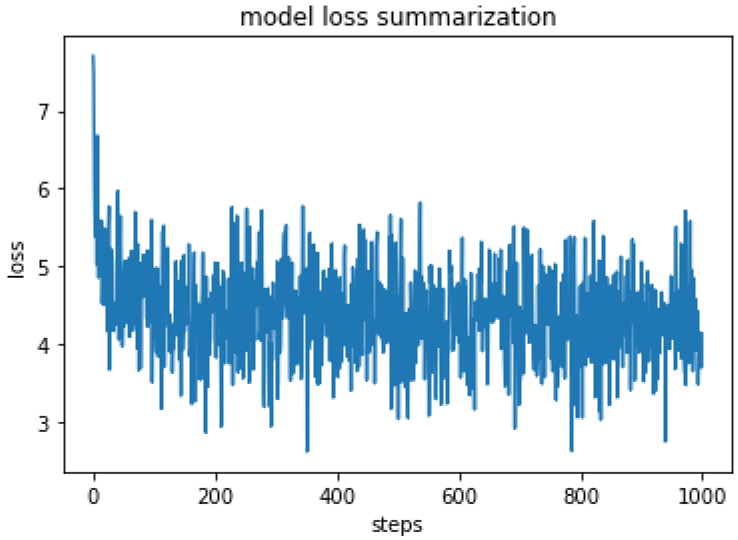
Task 2: QA

test rouge score: 0.339282986

train loss (avg); 4.381061132907868

vall loss (avg): 4.212564618110656

Graph:



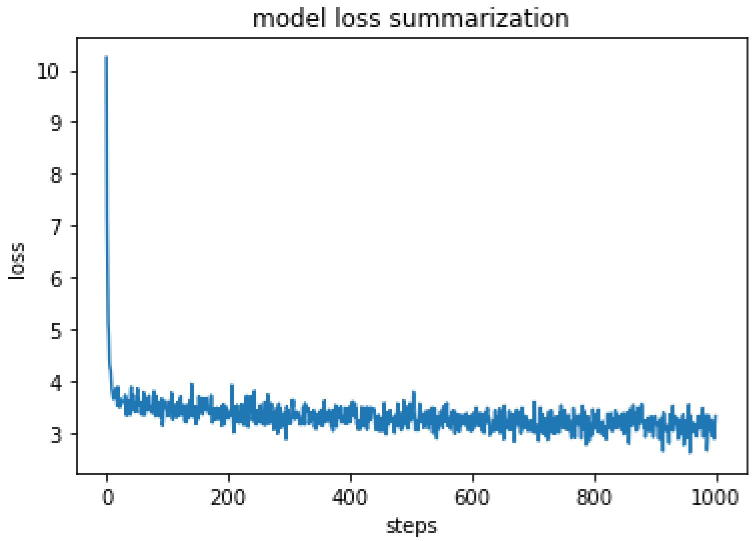
Task 3: translation

test rouge score: 0.5340372390605035

train loss (avg); 4.212564618110656

vall loss (avg): 4.381061132907868

Graph:


model loss summarization
loss
steps