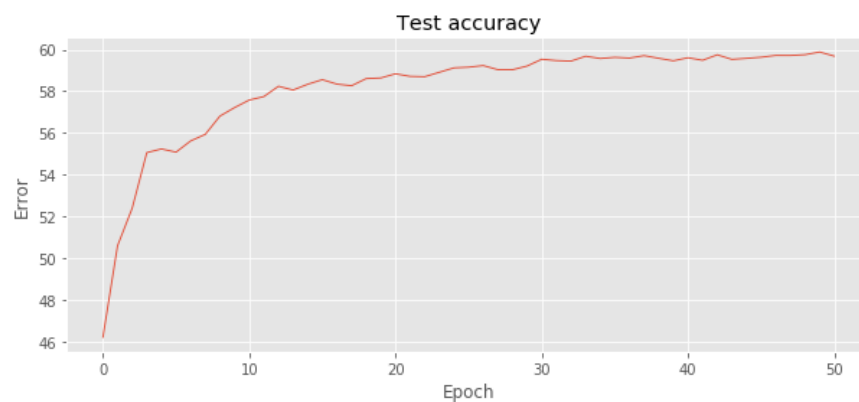


## Short answer questions

1. Plots of training/testing error, testing accuracy and also training/testing perplexity per epoch.

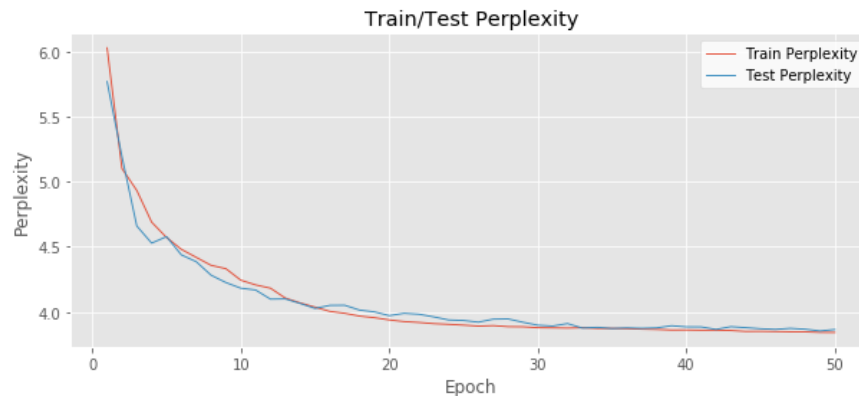
batch size	sequence length	total epochs	learning rate	feature size	optimizer
512	100	50	$0.001 \times 0.25^{ep//6}$	512	Adam()

```
HarryPotterNet(  
  (encoder): Embedding(90, 512)  
  (gru_1): GRU(512, 512, batch_first=True)  
  (decoder): Linear(in_features=512, out_features=90, bias=True)  
)
```



2. What was your final test accuracy? What was your final test perplexity?

The final results of the RNN model in Part 1 are **testing accuracy being 60.54%** and **testing perplexity being 3.9125**.



**3. What was your favorite sentence generated via each of the sampling methods? What was the prompt you gave to generate that sentence?**

The prompt I gave to generate sentence is *"Harry Potter, you are a ..."*.

**Max:** *"Harry Potter, you are a strange things and the strange thing they were still still still still staring at the trees are started to the first time the corner ..."*

**Sample:** *"Harry Potter, you are a surprise. Harry wore I make him all oaf your done. Dult about what is fresh!," said Professor suddenly pleasley, looking, at Harry ... "*

**Beam:** *"Harry Potter, you are a little right," said Hermione, looking around at Harry, who had never heard that he would have been through that it's nothing with you ... "*

**4. Which sampling method seemed to generate the best results? Why do you think that is?**

**Beam Search Sampling Method** generate the best sentence in our work. And the reason behind it is probably very trivial that beam search sampling method pick the best result by keeping a running score of the likelihood of each sequence among all possible sequences instead of giving the maximum possible value or random sampled value which are character-wise.

**5. Different Temperature**

**Which produces the best outputs?**

Given the value of **temperature within the range of 0.5 to 1.0** produce very readable outputs. While value above 1 generate stuff that look like garbled sequences and consist lots of made-up words.

Here are some example outputs with different temperature and same sampling method:

**Temperature=0:** *"Harry Potter, you are a stranger of the counter of the counter of the*

*counter of the counter of the counter of the counter of the counter ..."*

**Temperature=0.5:** *"Harry Potter, you are a room with his way on the stuff of the states when the look of player of contact of front of the door at the door. She was standing and start shot ..."*

**Temperature=1:** *"Harry Potter, you are a little right," said Hermione, looking around at Harry, who had never heard that he would have been through that it's nothing with you ..."*

**Temperature=2:** *"Harry Potter, you are a V(An.F owveNueapW! Sh9D!" ow?Chand." andzeded, "Jupiny?eoplaievol; poyketebse5ul you'a QuYcY tha5, isqumGvide ... KC: SqRE! ..."*

**Best as in made the most sense, your favorite, or funniest, doesn't really matter how you decide.**

*"Harry Potter, you are a matter of to all man!" Harry asked out a juicy ground. "The death better who haven't go anything, the middle or shocks, if you can five it?" It alto hit his throat other from out of place he was a suble-form with steps up on the school ..."*

**What does a temperature of 0 do? What does a temperature of  $0 < \text{temp} < 1$  do? What does a temperature of 1 do? What does a temperature of above 1 do? What would a negative temperature do (assuming the code allowed for negative temperature)?**

From the code in the inference() in our model, it was observed that temperature represents the value of how much to divide the logits by before computing the softmax for predictions. Therefore if given a temperature of 0, the model will be more likely to sampled from a more conservative output selections, which is actually identical to max sampling method.

When given a temperature between the value 0 and 1, the originally larger logits will be also larger after softmax due to the effect of dividing by the temperature, making the generation of the sequence somehow different from each other.

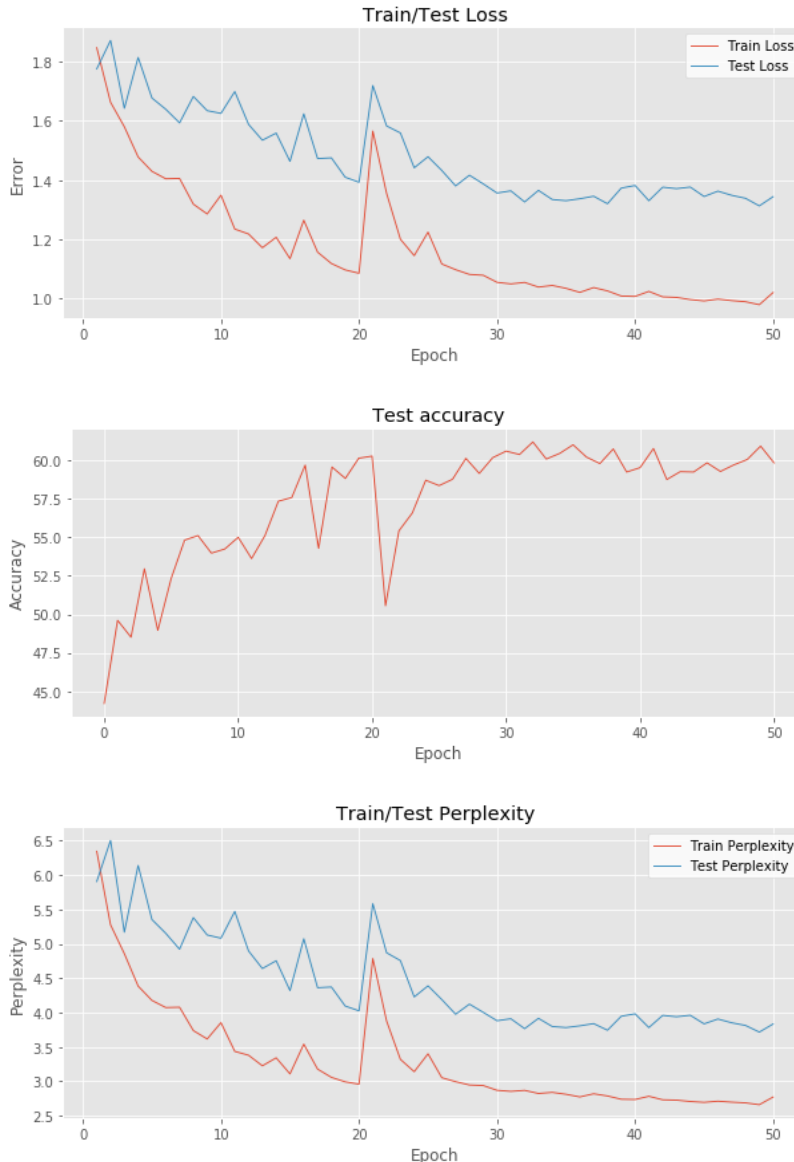
When given a temperature of 1, we do nothing to the logits!

When given a temperature above 1, the model will be more likely to sampled from a more diverse output selections. However, most of them resulted in meaningless words or sequences.

Given a temperature of negative value



Even though that the testing accuracy is the highest among all experiments, the generated sequences barely make any sense.



### 3. LSTM

The architecture replacing the GRU cells with LSTM cells resulted in similar result with the previous one, but converge a little faster. The **best testing accuracy is 59.654%** and **best testing perplexity is 3.9087**.

The prompt I gave to generate sentence is *"Harry Potter, you are a ..."* with the temperature and beam width chosen to be 0.8 and 10 respectively.  
**Max:** *Harry Potter, you are a good light of the more to see the door of the more to the stairs and started at the stairs and started at Harry ...*

```
HarryPotterLSTMNet(  
  (encoder): Embedding(90, 1024)  
  (lstm): LSTM(1024, 1024, batch_first=True)  
  (decoder): Linear(in_features=1024, out_features=90, bias=True)  
)
```

**Sample:** *Harry Potter, you are a few me to see that one to go to hear the world back on the prophecy from his way on the forest more, holding a strange gall stone ...*

**Beam:** *Harry Potter, you are a single of the morning of the tore you could see a burn, you know, how she was still be a new of them are more father ...*

