

For the extra credit tasks, I added some features and finetune to get the best alpha and rank.

1. For the feature engineering, I tried to add the length features into the model like what I did in the previous homework.

```
lrun = len(run)
lanswer = len(answer)
example["text"] = run + " [SEP] " + answer
example["label"] = correct
example["length of run"] = lrun
example["Length of answer"] = lanswer
dataset.append(example)
```

The first length is the length of the run (the part of question that has already been shown to the player) and the length of the answer. I also tested other features like 'category' and 'the ratio between the length of run and question'. Their performance under the default alpha=1 and rank=16 are shown in the following:

Features	Acc
Text only	0.6380208333333334
Text+L(run)+L(answer)	0.6692708333333334
Text+L(run)+L(answer)+Ratio	0.6597222222222222
Text+L(run)+L(answer)+Ratio+Category	0.6684027777777778

Thus, I finally choose the features combo of text + length_run + length_answer.

2. For the parameter finetuning part, I tried a rough CV with a group of rank and alpha.

alpha	rank	acc
1	32	0.6935671279180311
1	24	0.6657986111111112
1	16	0.6753472222222222
1	8	0.6948346889194043
1	4	0.6692708333333334
1.2	16	0.6605902777777778
0.8	16	0.6579861111111112
0.6	16	0.6753472222222222

Among them the best alpha and rank are 1 and 8. Though rank = 16 or 32 give similar accuracy, the lower rank of 8 further helps save some computational cost so it should be the best choice.