1 n-gram model

There are the following perplexities that I got. According to the TA, if the key did not exist in the model, then the perplexity would be considered to be infinity. Hopefully this is correct.

	train	dev	test
unigram	976.5	892.2	896.4
bigram	77.07	inf	inf
trigram	7.8	inf	inf

2 additional smoothing

Here are the tables of perplexities that I got with the following smoothings

	a=1			a=3			a=5		
	train	dev	test	train	dev	test	train	dev	test
unigram	993.2	908.8	896.4	1029.4	943.9	948.2	1067.3	980.4	984.6
bigram	1442.3	inf	inf	2666.8	inf	inf	3497.7	inf	inf
trigram	6244.8	inf	inf	10803.3	inf	inf	13005.2	inf	inf

Im not sure if the way I am doing smoothing, but with what I have, it appears that as I increase the alpha value, the higher my perplexities, so I think that the most viable value of alpha would be 1, simply based off the values that I got. (I might be very wrong)

3 linear interpolation

Note: For cases where the token did not exist in the bigram or trigram models, I skipped the word and moved on to the next token. If I had done it the other way all the perplexities for the bigram and trigram models would end up as infinity and I was not sure if it was correct.

Here are the following perplexities I got with linear interpolation. The three number on the leftmost column are the weights for unigram, bigram and trigram respectively. These are the perplexities that I got after cutting the training data in half and using that to fit my models.

lamba	train	dev	test
0.1, 0.3, 0.6	11.15	18.18	18.09
0.2, 0.2, 0.6	11.5	19.28	19.18
0.2, 0.4, 0.2	24.36	31.4	31.3

In my case it decreased the perplexities by a little bit. I assume this is because it had cut a lot of words that would initially have been considered as <UNK> which allowed more emphasis on the words that didn't count as <UNK>(more common words).

lamba	train	dev	test
0.1, 0.3, 0.6	10.78	16.08	15.82
0.2, 0.2, 0.6	11.2	17.04	16.77
0.2, 0.4, 0.2	22.07	27.35	26.96

lamba	train	dev	test
0.1, 0.3, 0.6	12.05	18.83	18.3
0.2, 0.2, 0.6	12.46	19.46	19.37
0.2, 0.4, 0.4	15.76	21.69	21.62

I tried it with my code and it seemed to have increased the perplexities. I think because words that appear between 3-4 times would be considered somewhat common, or at least might be important to certain sentences. However, I do think that this is a case by case basis, and in this case it does increase the perplexities.