Report on option 3

- Results on perplexity of the tested sentences:

	Word-level perplexity	Character-level perplexity
Sentence 1	3999.19880193086	5.778323524446959
Sentence 2	3531.5221654602433	6.507536673952857
Sentence 3	3086.9514322102805	6.599483210135334
Sentence 4	3930.5522718334755	6.825093373331272
Sentence 5	2408.257058952307	5.837556411630338
Sentence 6	2790.716363546713	6.027649035119668
Sentence 7	3709.3291645900013	6.24131688938205
Sentence 8	3351.1061060474276	6.8882875119618
Sentence 9	3945.6372143326985	7.478164133306564
Sentence 10	3908.3685319189794	6.1335773379296

- Description of my approach and analysis:

For option 3, I built a character-level 3-gram model based on provided dataset. During coding, I treated each character (including blank and punctuation) as a word and used the same approach of word-level 3-gram model construction for the purpose of word-level and character-level models comparison. For unknown characters, I used the “<unk>” technique: from the training data, I replaced any character with a frequency of 1 with the <unk> token and used the counts associated with <unk> anytime encountered an unknown character. For smoothing, I also used add-1 smoothing right after the <unk> pre-processing.

From the results table, we can clearly see that character-level model has much less perplexity values than word-level model. However, the pure perplexity values cannot be considered as a strong support to the argument that character-level model has better performance than word-level model. This is because the size of the word-level vocabulary is around 5700 while the character-level vocabulary is only around 70 (only 52 English letters, blank and punctuation). When applied to the smoothing as part of the denominator, vocabulary size plays a very important role in influencing the absolute perplexity values. In this case, in order to compare the word-level and character-level models in more reasonable ways, I propose a baseline model where perplexity is calculated based on random guess (random guess is the worst case), and both word-level and character-level models will compare to this baseline model to see which one has better performance. Towards this goal, I calculated the perplexity of baseline model first. Since baseline model is totally based on random guess, the probability will always be 1/V where V denotes the vocabulary size. Thus, the perplexity will be V. For word-level model, when compared to the baseline (worst case), the ratio of word-level perplexity (I use the mean of all the ten sentences' perplexity values) to baseline perplexity is around 0.67 while the character-level is around 0.09. This result can more objectively support the argument that character-level model has better performance than the word-level model.

This analysis result that character-level model has better performance can be partly explained through the fact that character-level model tend to consider grammar, semantic issues less. Currently, more SOTA research also tended to use character-level model, indicating the reliability of our analysis result.