ANALYSIS FOR P2-Markov
1. Runtime Complexities for BaseMarkov Implementation and their theoretical Justifications

setTraining() – O (T) is the expected complexity.
Given that T denotes the length/number of words of the training text, since we know that the setTraining method uses the .split method, in what is effectively an advanced for loop to split the training text of T elements , it's complexity is O(T).

getRandomText() – O(N*T) is the expected complexity.
The getRandom text method calls the nextWord method in an if loop which exists in a for loop. The nextWord method calls the getNextWord method which also calls the getFollows method. The getNextWord method has complexity O(1) for each time it is run since it only checks a condition. Within the helper method of the getRandomText method, this runs for N times and thus has O(N) complexity due to the for loop. But the getFollows method runs for T times, so a complexity of O(T) and is contained within the getRandomText method in compiling, thus the complexity is effectively O(N*T).

Experimental Justification -

Data File	T	N	Training Time	Generating Time
alice.txt	28196	100	0.015	0.128
alice.txt	28196	200	0.029	0.666
alice.txt	28196	12000	0.027	6.597
hawthorne.txt	85754	200	0.067	2.088
hawthorne.txt	85754	400	0.072	3.934
hawthorne.txt	85754	8000	0.061	71.890
kjv10.txt	823135	100	0.181	5.178
kjv10.txt	823135	400	0.140	13.137
kjv10.txt	823135	1000	0.312	81.426
romeo.txt	25788	100	0.034	0.248
romeo.txt	25788	200	0.042	0.619
romeo.txt	25788	400	0.029	1.248

Conclusion:
For setTraining time – O(T)
The experimental data is consistent with the theoretical predictions to some extent but not entirely. This is seen in how there exists some level of consistency in the training time for the same texts regardless of the number of words being generated. For example, for hawthorne.txt, regardless of the number of words being generated, the training time is approximately 0.07s. However, with changes in T, there is approximate change in the training time. Alice.txt and romeo.txt, have approximately same training time due to approximately equal number of words in both texts. Where there is a significant change in the number of words for instance between hawthorne and alice, we see a similar growth in training times.

For the getRandomText time – O(NT)
AS the value of N changes for the same T, we see a proportional growth in the generating time. For instance for romeo.txt, when N changes from 200 to 400, the time changes by a factor of 2 from 0.6s to 1.2s. Thus this is also conclusive with our predictions as the data above shows

2. Runtime Complexities for HashMarkov Implementation and their Theoretical Justifications

setTraining()-O(T)
The setTraining for HashMarkov has a for loop that iterates through the words of myWords and since there are T elements, its complexity is O(T) in the for loop which runs for T times. The .split method also runs for O(T) complexity and thus the total complexity for the method is O(T)

getRandomText()- O(N)
For a hashmap, the getNextWord has a complexity of O (1) since it doesn't iterate through the text each time but simply gets it with a complexity of O(1). Thus the helper method has a complexity of O(1) but since it runs within a getRandomText loop that goes on for N times, the complexity is O(N). And this is different from baseMarkov since HashMaps implement in a different way.

Experimental Justification -

Data File	T	N	Training Time	Generating Time
alice.txt	28196	100	0.077	0.000
alice.txt	28196	200	0.071	0.001
alice.txt	28196	12000	0.071	0.004
hawthorne.txt	85754	200	0.244	0.001
hawthorne.txt	85754	400	0.269	0.002
hawthorne.txt	85754	8000	0.528	0.022
kjv10.txt	823135	100	1.810	0.001
kjv10.txt	823135	400	2.219	0.002
kjv10.txt	823135	1000	2.179	0.004
romeo.txt	25788	100	0.070	0.000
romeo.txt	25788	200	0.077	0.001
romeo.txt	25788	400	0.083	0.001

The training time just as with the first experiment does not show any direct correlation with the value of N, and appears to be fairly consistent for the same values of T. However for generating time, we notice that overall the HashMarkov implementation outperforms the BaseMarkov implementation even though the BaseMarkov appears to do better for training time. For Hashmarkov, generation time reflects almost instant results, which are proportional to the number of words being generated.

3.

OpenAI's mission to ensure that artificial general intelligence (AGI) benefits all of humanity could be generally viewed positively. The goal of making AI safe and accessible aligns with the idea of avoiding concentration of power and ensuring widespread benefits. However, the notion of highly autonomous systems outperforming humans at most economically valuable work raises concerns about potential societal impacts. The benefits of such systems would depend on how they are implemented, regulated, and integrated into society. Ensuring that the development of AGI considers ethical, social, and economic implications will be crucial for achieving positive outcomes for humanity.

The debate around open-source code in AI/ML is multifaceted. Open source has several advantages, such as fostering collaboration, transparency, and community-driven improvement. It allows researchers and developers worldwide to contribute, understand, and build upon existing models. However, there are also concerns about intellectual property, commercial interests, and the potential misuse of powerful technologies. Striking a balance is essential. While open-source initiatives can accelerate progress and democratize access to AI/ML, it's understandable that certain models or technologies may be initially proprietary to ensure responsible development, gradual deployment, and consideration of safety and ethical aspects. In the context of the article about GPT-J being an open-source alternative to GPT-3, providing open-source alternatives can address accessibility concerns and promote innovation in the field. However, the decision to open-source code should consider various factors, including the nature of the technology, potential risks, and the overall impact on society.