Part 5: Analysis and Reflection

Quality Comparison: Improvement Across Approximation Levels

The quality of the generated text improves as the model order increases from zero-order to third-order.
Each higher level provides the model with a larger context window, allowing it to make better predictions and produce text that is more grammatically consistent.

- **Zero-Order (Random Generation)**
  At this level, characters or words are chosen entirely at random.
  Example (*Austen*):

  > "iqt dhq afeean n rrd t sdteh yta…"
  > The output is meaningless and does not contain recognizable words or structure. It only reflects basic frequency counts of individual letters.

- **First-Order (Unigram Model)**
  The first-order model selects words based on how frequently they appear in the original text.
  Example:

  > "the of and to she it in was had her be not…"
  > While it generates real words, there is no logical sequence or sentence structure. The text reads as a list of high-frequency words without coherence.

- **Second-Order (Bigram Model)**
  In this model, each token depends on the previous one.
  Example:

  > "she was not a woman of sense and her temper was very…"
  > Short, grammatically plausible fragments begin to appear. However, the model still loses context quickly and fails to produce complete sentences.

- **Third-Order (Trigram Model)**
  The third-order model considers two previous tokens, which provides enough context for short, fluent phrases.
  Example:

  > "and they took place they must have occasion to her of georgiana's delight in her interest…"
  > The result is still semantically weak, but it resembles natural English more closely. The phrasing and structure are recognizably similar to Austen's style.

Overall, increasing the model order improves the grammatical consistency and local coherence of the generated text. The transition from bigram to trigram models shows the largest improvement, as the additional context allows the generator to produce text that follows realistic syntactic patterns.

Author Distinctiveness: Recognizing Style Across Levels

At the lower approximation levels (zero- and first-order), the generated text does not reflect any author-specific style.
The output consists mainly of random words or letters arranged by frequency, so it could have originated from any English text.

By the second-order (bigram) model, limited stylistic elements begin to appear. For example, the Austen model starts producing fragments such as "she was not" or "of her," which reflect her frequent use of polite and descriptive phrasing.
The Twain model occasionally generates contractions such as "ain't" or "warn't," showing hints of his informal, conversational tone.
Doyle's bigram output, while still inconsistent, contains recurring logical connectors like "therefore" or "because," which are common in his deductive writing style.

At the third-order (trigram) level, author distinctiveness becomes much clearer.
The Austen model tends to form longer and more structured sentences resembling classical prose.
Twain's model consistently generates short, colloquial phrases that sound conversational.
Doyle's text shows a more direct, analytic structure that fits his detective narrative style.

The trigram models are the first level where each author's voice becomes clearly recognizable. The higher-order context allows the model to capture not just word frequency, but also habitual phrasing patterns unique to each writer.

Anchor Word Integration: Challenges and Solutions

Integrating specific anchor words into the generated text was challenging because it often disrupted the flow of the sentence.
When an anchor word was inserted manually, the surrounding words did not align grammatically, and the model would produce unnatural transitions.

To address this, I used a simple seeding approach.
Instead of forcing the anchor word into the middle of a sentence, the model began generation from a context that included the chosen word or phrase.
For example, if the anchor was "love" (for Austen) or "Watson" (for Doyle), the text generation started with that token and then continued normally based on the learned n-gram probabilities.

This approach worked better because it allowed the model to generate words naturally around the anchor, maintaining a smoother sentence structure.
It still gave the desired word prominence without breaking syntax.
The main limitation was that the model could not always keep the topic centered on the anchor for long, since the generation process remains probabilistic and not goal-directed.

Modern Connections: N-Grams vs. Neural Language Models

The n-gram models used in this project represent an early stage of language modeling, where text generation is based purely on local statistical relationships.
Modern neural language models, such as GPT or BERT, extend this same idea but operate on a much larger scale and with a deeper understanding of context.

Feature	N-Gram Models	Neural Language Models
Context Window	Fixed (limited to n-1 previous tokens)	Variable and large, often spanning hundreds or thousands of tokens

Feature	N-Gram Models	Neural Language Models
Memory	None beyond immediate context	Long-range context through attention mechanisms
Representation	Frequency-based counts	Continuous embeddings that capture semantic meaning
Limitations	Data sparsity, limited coherence, rigid context	Computational cost, requires large datasets and hardware

N-gram models can only capture short-term dependencies, which limits how coherent their output can be.
For example, a trigram model can mimic the surface style of an author but loses track of meaning after a few sentences.
Neural models overcome this by learning vector representations of words and by using attention layers to connect ideas across longer stretches of text.

Modern systems like GPT can be seen as the logical continuation of Shannon’s idea.
They still model language as a probabilistic process but do so with far greater capacity to understand context, syntax, and even semantics.

Shannon’s Insight: Language as a Stochastic Process

Claude Shannon’s main contribution was the idea that language is a **stochastic process**—it is governed by conditional probabilities.
Each symbol or word depends on the ones that come before it, and this dependence can be modeled mathematically.

This project directly demonstrates that concept.

- Zero-order models produce random noise, representing maximum entropy where there is no predictability.
- As the order increases, each model captures more contextual information, reducing uncertainty and producing text that better resembles real language.

In information-theoretic terms, increasing the model order lowers the conditional entropy *H(X|context)*, since the model becomes more confident about what comes next.
This progression—from randomness to structured text—illustrates Shannon’s idea that communication can be understood through probabilities rather than strict rules.

The experiment also connects to modern AI by showing how today’s language models still rely on Shannon’s principles.
Although neural networks operate at a much larger scale, they are built on the same foundation: predicting the next token based on prior context.
This assignment therefore serves as a small-scale, practical example of how Shannon’s view of language as a probabilistic system remains central to natural language processing today.

Summary Reflection

This assignment demonstrates, in a simplified way, how Shannon’s 1948 ideas about information and probability apply to language.
By building models of increasing order, it becomes clear that adding context gradually improves the structure and coherence of generated text.
Even with simple n-gram models, the results show that predictability and repetition play an important role in how language conveys meaning.

The exercise also highlights how these early statistical methods form the basis of modern language modeling.
Although contemporary neural models are far more complex, they still rely on the same principle of predicting the next token based on prior context.
Overall, the project provides a practical understanding of how language can be treated as a probabilistic system, linking Shannon’s theoretical insights to current approaches in artificial intelligence.