# Understanding the Parameters

* **Temperature (The Randomness Dial):** This parameter controls how much randomness is introduced into the word selection process.
  + Low temperature (e.g. 0.0 – 0.3): The model becomes more deterministic and conservative. It focused, predictable, and often repetitive.
  + High temperature (e.g. 0.8 – 1.2): The model takes more risks. It increases the chance of selecting less likely words, leading to more diverse, creative, and sometimes surprising outputs. However, a very high temperature can also resul in nonsense or hallucinations.
* **Top -p / Nucleus Sampling (The Vocabulary Filter):** This parameter controls the diversity of the output by limiting the pool of words the model can choose from. It works by creating a “nucleus” of the most probable words.
  + Low Top-p (e.g. 0.1 – 0.3): The model considers only a very small set of the most likely words. For example, if top\_p is 0.1, the model only looks at the top words combined probability is 10%. This makes the output very narrow and safe.
  + High Top\_p (e.g. 0.9 – 1.0): The model considers a much larger and more diverse set of words. A top\_p of 1.0 means the model can choose from any word in its vocabulary that has a non-zero probability.
* **Impact on Latency:** While Temperature ant Top-p are primarily used to control the creative output of the model, they can also have a minor impact on latency (the time it takes to generate a response.)
  + **Temperature:** This parameter generally has a negligible impact on latency. The calculation is a simple mathematical adjustment applied to the final probability scores and does not add significant computational overhead.
  + **Top-p:** This parameter can have a small effect. A very low top-p value (e.g., 0.1) might slightly reduce latency because the model only needs to process and sample from a very small set of tokens. Conversely, a high top-p (e.g., 1.0) can marginally increase latency as the model has a much larger pool of potential words to consider before making its selection.
  + For most applications, this difference in performance is minimal, but it can be a factor in highly time-sensitive or optimized systems.

# Four Common Scenarios

Here are four different situations combining these parameters, which produce distinct behaviours.

## Low Temperature / High Top-p: The Reliable Assistant

* **Settings:** temperature = 0.2, top\_p = 0.9 (This is similar to your script's configuration)
* **Behavior:** This combination produces highly coherent and predictable text. The **low temperature** ensures the model sticks to the most logical and probable words, while the **high top-p** provides just enough flexibility to prevent the text from being overly robotic or repetitive. It's a safe and balanced approach.
* **Best Use Cases:** Factual question-answering, document summarization, code generation, and Retrieval-Augmented Generation (RAG) systems where you want a reliable answer based on provided context.

## High Temperature / High Top-p: The Creative Brainstormer

* Settings: temperature = 0.9, top\_p = 0.9
* **Behaviour:** This is the most creative and adventurous setting. The **high temperature** encourages the model to try unusual word choices, and the **high top-p** gives it a vast vocabulary to choose from. The output will be highly diverse, novel, and sometimes unexpected. This is where you'll find poetry, story ideas, and unique marketing slogans. The trade-off is a higher risk of the output becoming nonsensical or factually incorrect.
* **Best Use Cases:** Creative writing, brainstorming ideas, composing marketing copy, and generating dialogue for characters.

## Low Temperature / Low Top-p: The Focused Expert

* **Settings:** temperature = 0.1, top\_p = 0.1
* **Behaviour:** This is the most constrained and deterministic setting. Both parameters force the model to choose from a very small set of the most obvious words. The output will be extremely consistent, focused, and highly repetitive. It will have almost no variation between generations for the same prompt.
* **Best Use Cases:** Classification tasks, data extraction, or any situation where you need the most predictable and repeatable single answer possible.

## High Temperature / Low Top-p: The Unpredictable Gambler

* Settings: temperature = 0.9, top\_p = 0.2
* **Behavior:** This is an unusual and often chaotic combination. The **low top-p** creates a small, safe pool of likely words, but the **high temperature** then forces the model to make a random choice from within that tiny pool. The result can be strange and incoherent because the model is being forced to be random within a very restricted set of options. It's like telling someone to "be creative!" but only letting them use the 100 most common words.
* **Best Use Cases:** This combination is **generally not recommended** for practical applications. It's mostly used by researchers to experiment with the model's behavior under strange constraints.

# Analysis of Results

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 1: Low Temperature / High Top-p Result

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2: High Temperature / High Top-p Result

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 3: Low Temperature / Low Top-p Result

A computer code on a dark background

AI-generated content may be incorrect.

Figure 4: High Temperature / Low Top-p Result

The key takeaway from comparing the four output files is that a well-structured, context-based prompt (like the one used in your RAG system) significantly narrows the model's potential responses, making it less sensitive to creative settings.

## Consistency in Factual Answers

For prompts where a direct answer was available in the knowledge base, the output was nearly identical across all four settings.

* **Mobile Crane MC750:** The answer was exactly the same in all logs: "the maximum lifting capacity of the MC750 Large Mobile Crane is 750 metric tons / 826 short tons".
* **Dump Truck DT1000:** The answer was identical in three of the four logs: "the maximum payload capacity of the DT1000 Large Mining Dump Truck is 363 metric tons or 400 short tons".

This demonstrates that when the context provides a clear, unambiguous fact, the model will reliably extract it regardless of the creativity settings.

## Stylistic Variations

The most noticeable difference appeared in the **High Temperature / High Top-p** setting ("Creative Brainstormer"), where the phrasing was slightly more conversational and varied.

* **Bulldozer vs. Excavator Prompt:** While all four logs correctly stated that information on the "Excavator X950" was missing , the creative setting's response was phrased differently and concluded with "Therefore, I cannot answer the question based solely on the provided context". The other three settings were more direct.
* **Dump Truck Prompt:** This was the clearest example of a stylistic change. Instead of just providing the answer, the creative setting quoted the source material and explained its reasoning: "The context states that the DT1000 Large Mining Dump Truck has a 'Payload Capacity: 363 metric tons / 400 short tons'. This directly answers the question...".

# Conclusion

For a fact-based Q&A system built on Retrieval-Augmented Generation (RAG), the results show that a **low temperature** setting is most effective. It consistently provides accurate and concise answers by sticking closely to the provided context. The high-temperature settings did not yield better answers and introduced slight verbosity without adding factual value. The strong constraint of the RAG prompt ("Answer the question using ONLY the following context") successfully prevented the model from hallucinating or providing incorrect information, even in its most creative configuration.