Temperature and Top_P: How They Control AI Responses

Author: Ansar Rezaei
Project: AWS Bedrock Knowledge Base with Heavy Machinery Document Querying System
Purpose: This document explains how I configured temperature and top_p parameters in my RAG (Retrieval-Augmented Generation) application for querying heavy machinery specifications.

When working with AI language models, I can control how the model generates responses using two important parameters: temperature and top_p. These settings help me get the right kind of answer for my needs. I can choose between creative writing or precise technical information.

In my project, I built a Streamlit application that uses Amazon Bedrock Knowledge Base connected to Aurora PostgreSQL Serverless. The application allows users to ask questions about heavy machinery equipment (excavators, bulldozers, cranes, etc.) and retrieves accurate information from uploaded specification documents. Understanding and configuring temperature and top_p correctly was crucial for getting accurate, technical answers rather than creative or unpredictable responses.

Temperature

Temperature controls how creative or predictable the AI answers are. Think of it like this: when temperature is 0, the AI always picks the most likely word next. This gives very consistent and focused answers. This is perfect for technical information. When I increase the temperature (like from 0 to 1), the AI becomes more creative and less predictable. It starts choosing words that are less common but more interesting.

For example, if I ask about an excavator's engine at temperature 0.3, the AI will give a clear, technical answer: "The excavator uses a diesel engine with 250 horsepower." But at temperature 0.9, it might say something more creative: "This powerful machine roars to life with a robust diesel heart, pumping out 250 horses of pure mechanical muscle."

In my heavy machinery application, I keep temperature low (around 0.3) because I want accurate and consistent answers about equipment. I don't need creative writing. I need facts!

Top_P

Top_P controls which words the AI can choose from. It's like a filter that says "only consider the most likely words." When top_p is 0.1, the AI only looks at the top 10% most likely words. This keeps the answer focused and precise. When top_p is higher (like 0.9), the AI can choose from many more words. This makes answers more varied but sometimes less accurate.

Here's a simple way to understand it: imagine the AI has a list of 100 possible words to use next. With top_p at 0.1, it only looks at the top 10 most likely words. With top_p at 0.9, it looks at the top 90 words. More choices mean more variety, but also more chance of using words that don't fit perfectly.

In my application, I use a low top_p value (0.1) because I want the AI to stay focused on technical terms about heavy machinery. I don't want it to use random words that don't make sense for my topic.

How They Work Together

Temperature and top_p work together to control the AI responses. Temperature decides how creative the AI is with word choices. Top_p limits which words the AI can even consider. In my heavy machinery application, I use both settings low (temperature 0.3, top_p 0.1) to get accurate, focused, and technical answers every time. This combination is perfect for answering questions about equipment specifications. For this use case, accuracy matters more than creativity.

Different Settings for Different Tasks

In my application, I actually use different temperature and top_p settings for different purposes. When I need to classify user prompts (to check if they're asking about heavy machinery), I use the strictest settings possible: temperature 0 and top_p 0.1. This makes the classification always consistent. The same question will always get the same category. But when generating answers to user questions, I use temperature 0.3 and top_p 0.1. This gives slightly more natural language while still keeping answers accurate and focused. The key is matching the settings to the task: strict settings for classification, slightly relaxed settings for natural conversation.

What I Learned from Testing

I ran 15 different tests to understand how these parameters work in practice. Here is what I observed:

Temperature Effects (Comparing 0.3 vs 0.7): When I asked "Describe the main features of the X950 excavator" with temperature 0.3, I got direct and technical answers. When I increased temperature to 0.7, the response was still well-organized with 6 numbered main features. The information stayed accurate and factual. The structure was clear with categories like "Proven Design Philosophy," "High Efficiency," and "Operator-Focused Cab Design." This shows that even at higher temperature, the AI maintains good organization for technical content. However, I prefer temperature 0.3 for my use case because it ensures maximum consistency and focus on facts.

Top_P Effects (Comparing 0.1 vs 0.5): When I tested top_p at 0.1 versus 0.5, I asked about bulldozer safety features. With top_p 0.5, the response provided 7 detailed safety features in a numbered list. The language was clear and technical. The response included specific details like "ROPS/FOPS certified cab" and "Advanced onboard diagnostics system." The difference between 0.1 and 0.5 was subtle in this case. Both settings produced well-structured, factual responses. I keep top_p at 0.1 because it ensures the most focused word choices for technical terminology.

Knowledge Base Results (Comparing 3 vs 8 results): I tested asking about terrain capabilities with 3 results versus 8 results. With 3 results, I got focused answers from fewer sources. With 8 results, the sources section showed 8 different sources with confidence scores ranging from about 37% to 54%. This gave me more context from multiple equipment documents. I found that 3 results works well for specific questions about one piece of equipment. But 5-8 results works better for general questions or when comparing multiple machines, as it pulls information from more documents.

Model Comparison (Haiku vs Sonnet): I tested the same question "Explain the operating specifications of the MC750 mobile crane" with both Claude Haiku and Claude Sonnet models. Both models provided detailed, well-structured responses with bullet points covering lifting capacity, boom length, engine power, and fuel efficiency. Both maintained accuracy and good organization. This shows that both models work well for technical queries when using the same temperature (0.3) and top_p (0.1) settings.

Sources and Citations: The sources feature works well. When I asked about fuel type for the FL250 forklift, the system showed 2 sources with confidence scores (44.71% and 42.19%) and file paths to the PDF documents. This helps users verify where the information came from. The confidence scores help understand how relevant each source is to the question.

Prompt Validation: My validation system successfully blocked all inappropriate prompts. It correctly rejected questions about system architecture (Category A), toxic content (Category B), off-topic questions like weather (Category C), and prompt injection attempts (Category D). It also rejected prompts that were too short (less than 10 characters). Each rejection showed a clear, helpful message explaining why the prompt was rejected. This proves that using temperature 0 and top_p 0.1 for classification gives consistent and reliable results every time.