

Capstone Project Report

Project Overview:

In this project we are creating a text completion tool using the Gemini API in VS Code. The goal is to generate natural language completions based on various input prompts by using the API's large language model capabilities.

What is an API ?:

Application Programming Interface is what allows different applications and software tools to communicate with each other. This allows us to implement different tools like Gmail, Drive, OpenAI, etc .. in our own software and use them.

In this project I am using the Google Gemini API key through the Google AI studio app because it is free.

Key Parameters:

It is important to note that the generation of the prompts depends on key parameters which are:

- Temperature (k):
Controls the randomness of the generated text. A lower temperature (e.g., 0.2) makes the output more focused by favoring the most likely next tokens. A higher temperature (e.g., 0.8 or above) increases randomness and creativity by allowing less probable tokens to be selected.
- Max Tokens (max_output_tokens):
Sets the maximum number of tokens (words) the model will generate in the completion. It is effective when dealing with models that cost money to control the output. For example, a max token value of 100 means the model will generate at most 100 tokens.
- Top-p (top_p):
Controls diversity by limiting the next token choices to a subset whose probability mass is at least p. Lower top-p values reduce randomness by focusing on high-probability tokens, while higher values allow more diverse and creative outputs.

Prompting:

In this section we will be modifying the parameters and trying different prompts to assess the Gemini model.

Prompt 1: What day is it today ?

Parameters: k = 0.7, max_output_tokens = 300, top_p = 0.9

Output: Today is Friday, October 27th, 2023

Fares Yahmadi
06/11/2025

Observation: The response is clear and coherent. Even though it doesn't answer correctly, these parameters are good for a factual question that requires a concise, confident answer.

Prompt 2: As someone who has never been to Italy, describe how it feels and looks like to be in Rome.

Parameters: $k = 1$, $\text{max_output_tokens} = 150$, $\text{top_p} = 0.9$

Output: Imagine stepping off the plane and the air immediately feels different – warmer, drier, perhaps a little dusty. The scent hits you first: a heady mix of exhaust fumes, sun-baked stone, and something indefinably *Roman*, a blend of ancient history and modern life. The first impression is one of overwhelming scale and age. Buildings seem to lean in on each other, their ochre and terracotta hues softened by centuries of sun and weathering. You're constantly aware of layers of time – a crumbling Roman wall might be next to a Renaissance palace, which is itself adjacent to a modern apartment building. It's a visual tapestry woven from different eras.

The streets are a chaotic ballet of Vespas zipping between

Observations: Coherence: The description is vivid and flows naturally. The increase in K makes the description much more creative. Despite that, the structure remains logical and well-paced. However, the output seems to be interrupted at "a chaotic ballet of Vespas zipping between". This is due to limiting the max_tokens to 150.

Prompt 3: Write a professional email to a professor asking for an extension on a homework assignment.

Parameters: $k = 0.3$, $\text{max_output_tokens} = 100$, $\text{top_p} = 0.8$

Output: Subject: Request for Extension - [Your Name] - [Course Name] - [Assignment Name]

Dear Professor [Professor's last name],

I am writing to respectfully request an extension on the [Assignment Name] assignment, originally due on [Original Due Date]. I have encountered [Briefly and honestly explain your reason for needing an extension. Be specific but avoid excessive detail. Examples: an unexpected illness, a family emergency, a significant workload from another demanding course].

I

Observations: The low temperature and top_p values result in a well written and professional email. Coherence is great, and the model is answering the prompt. Creativity is very limited and the model seems to be interrupted and did not finish the email.

Prompt 4: Write a haiku about the internet.

Parameters: $k = 1.2$, $\text{max_output_tokens} = 30$, $\text{top_p} = 0.95$

Output: World wide web of words,
Connecting minds across space,
Knowledge at our touch.

Observations: The high temperature and top_p values allow for an abstract language. Coherence is preserved within the haiku format, and the result is rich. This output demonstrates the model's generative strength when creativity is emphasized.

Prompt 5: Explain what Transformers are and how they work to a 12 year old.

Parameters: k = 0.8, max_output_tokens = 100, top_p = 0.9

Output: Imagine you have a really big box of LEGOs, all mixed up. You want to build a spaceship, but finding the right pieces is a nightmare!

That's kind of what it's like for a computer trying to understand a sentence or a picture. The words or pixels are like the LEGOs – all jumbled up. Traditional computer programs would have to look at each LEGO (word or pixel) one by one, in order. This is slow and doesn't

Observations: The analogy with LEGOs is relatable and creative, which helps simplify a complex concept. The coherence is strong up until the cutoff. Like Prompt 2, the response is cut off mid-sentence due to the token limit being too low.

Conclusion:

Through these five prompts, I tested how different parameters affect the Gemini API's responses. Lower temperature values (e.g., k = 0.3) produced safe, structured, and predictable outputs which are ideal for formal or factual content. Higher temperatures (k = 1 or above) encouraged more creativity and description. Top-p controlled the diversity of word choices, while max_output_tokens directly influenced the completeness of responses. Overall, the experiment showed that tuning these parameters allows for better control over tone, coherence, and creativity depending on the task.