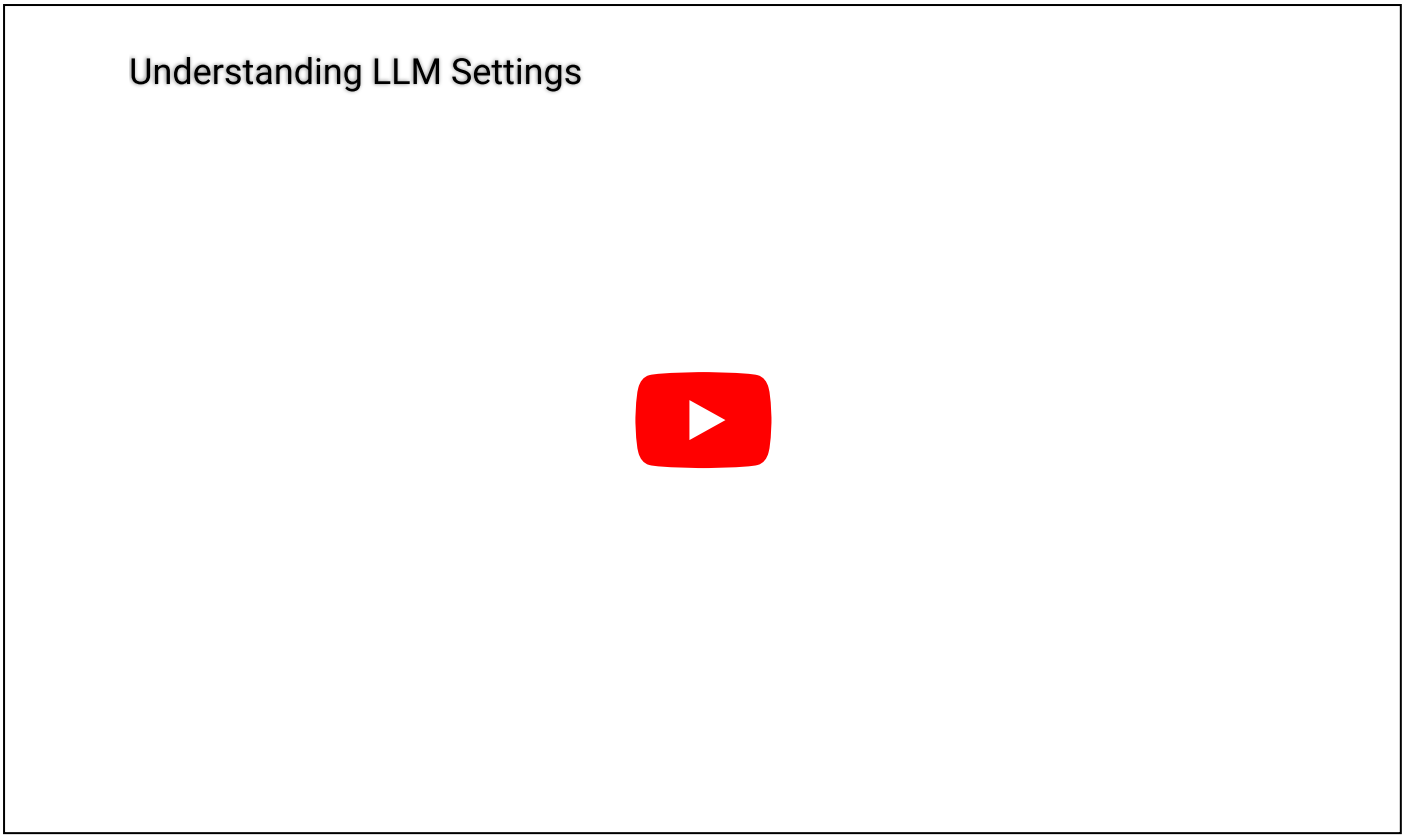
LLM Settings



When designing and testing prompts, you typically interact with the LLM via an API. You can configure a few parameters to get different results for your prompts. Tweaking these settings are important to improve reliability and desirability of responses and it takes a bit of experimentation to figure out the proper settings for your use cases. Below are the common settings you will come across when using different LLM providers:

Temperature - In short, the lower the `temperature`, the more deterministic the results in the sense that the highest probable next token is always picked. Increasing temperature could lead to more randomness, which encourages more diverse or creative outputs. You are essentially increasing the weights of the other possible tokens. In terms of application, you might want to use a lower temperature value for tasks like fact-based QA to encourage more factual and concise responses. For poem generation or other creative tasks, it might be beneficial to increase the temperature value.

Top P - A sampling technique with temperature, called nucleus sampling, where you can control how deterministic the model is. If you are looking for exact and factual answers keep this low. If you are looking for more diverse responses, increase to a higher value. If you use Top P it means that only the tokens comprising the `top_p`

probability mass are considered for responses, so a low `top_p` value selects the most confident responses. This means that a high `top_p` value will enable the model to look at more possible words, including less likely ones, leading to more diverse outputs.

The general recommendation is to alter temperature or Top P but not both.

Max Length - You can manage the number of tokens the model generates by adjusting the `max length`. Specifying a max length helps you prevent long or irrelevant responses and control costs.

Stop Sequences - A `stop sequence` is a string that stops the model from generating tokens. Specifying stop sequences is another way to control the length and structure of the model's response. For example, you can tell the model to generate lists that have no more than 10 items by adding "11" as a stop sequence.

Frequency Penalty - The `frequency penalty` applies a penalty on the next token proportional to how many times that token already appeared in the response and prompt. The higher the frequency penalty, the less likely a word will appear again. This setting reduces the repetition of words in the model's response by giving tokens that appear more a higher penalty.

Presence Penalty - The `presence penalty` also applies a penalty on repeated tokens but, unlike the frequency penalty, the penalty is the same for all repeated tokens. A token that appears twice and a token that appears 10 times are penalized the same. This setting prevents the model from repeating phrases too often in its response. If you want the model to generate diverse or creative text, you might want to use a higher presence penalty. Or, if you need the model to stay focused, try using a lower presence penalty.

Similar to `temperature` and `top_p`, the general recommendation is to alter the frequency or presence penalty but not both.

Before starting with some basic examples, keep in mind that your results may vary depending on the version of LLM you use.

🎓 Learn more about LLM settings in our new AI courses. Join now!

Last updated on January 7, 2025

Copyright © 2024 DAIR.AI