Chatbot report:

In a separate note book, Exercise a chatbot on different level, with different answer and different possibility.

Temperature: In the context of chatbot conversational dynamics, Experiencing the difference between temperature into different values (0.5 - 1.5) the higher the temperature is the more friendly the chatbot will be.

Top_K: When the computer is generating text, it looks at all the possible words it could use next. It has a list of these words, and each one has a probability - some are more likely than others.

Instead of just picking the single most likely word, the computer looks at the top 3, 5, or 10 most probable words on the list. It sorts them from most likely to least likely.

Then, the computer adjusts the probabilities of just those top words, so they add up to 100%. This means the less likely words at the bottom of the list don't get chosen.

Finally, the computer randomly picks one of those top words, but it chooses the more likely ones more often. This helps the computer generate text that flows better and makes more sense, rather than picking weird or random words.

It's a way for the computer to be a bit smarter about which words it uses, instead of just going with the single most obvious choice every time. It makes the computer-generated text sound more natural and coherent.

Top_P: these language model programs, when they're generating text, they've got this whole list of possible words that could come next. And each word has a probability, like how likely it is to show up.

Instead of just looking at the top few most probable words, this top-p method is a bit different. It sorts the words, adds up the probabilities from the top until it hits a certain threshold, and uses that subset.

The neat part is, the number of words it looks at can change based on how the probabilities are spread out. So sometimes it's the top 5, other times it might be more. It just grabs the words that add up to that threshold.

Then it randomly picks one of those words, but weights it by the probabilities. So the more likely options have a better shot.

The idea is to give the program a little more flexibility to generate more natural, contextual text. Rather than always looking at the same number of words, it can adapt based on the probability distribution.