The Pivot Word

Naïve Proof of Inference for Autoregressive Models

Vitaly Kleban, Brussels, 2024
@vkleban

Alice wants to run an Llama-3-8B model inference on a set of inputs but doesn't have enough computing power.

Bob has the necessary computing power and offered to help Alice by performing the inference for her.

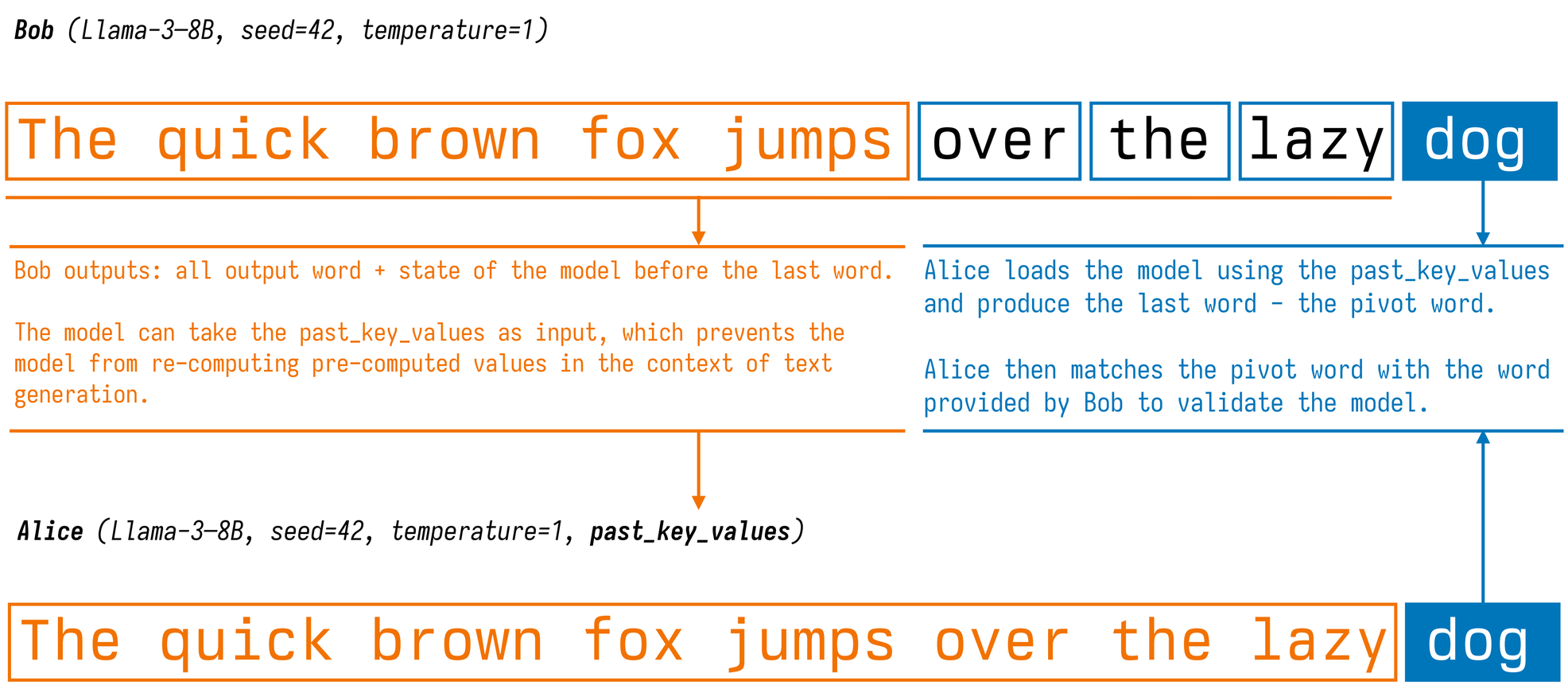
Now, **Alice** wants to verify that **Bob** actually used the specified model for the inference.

Alice needs to spend much less time verifying the input rather than doing the inference.

Bob *(Llama-3-8B, seed=42, temperature=1)*

The next token generated by the LLM is dependent on the sequence of tokens that have been provided as input up to that point and the state of the model (previously computed key/value attention pairs).

Alice *(Llama-3-8B, seed=42, temperature=1)*



- The model can take the *past_key_values* (for PyTorch) or *past* (for TF) as input, which is the previously computed key/value attention pairs. Using this (*past_key_values* or *past*) value prevents the model from re-computing pre-computed values in the context of text generation. For PyTorch, see *past_key_values* argument of the GPT2Model.forward() method, or for TF the *past* argument of the TFGPT2Model.call() method for more information on its usage.

Bob processes all the words and shares the results with Alice.

The final word, the pivot, will determine if Alice trusts Bob.

Bob also sends the attention values recorded just before generating the pivot word.

Alice gets the output, including the pivot word from Bob.

She loads the model with Bob's attention values and runs a one-word inference to see if it matches the pivot word.

Chat with me at @vkleban!

Send your pitch to vk@cyber.fund

Check out “salty embeddings” from the same author

https://github.com/vkleban/salty-embeddings