Differences of Executors

Differences between the executors

There're currently three kinds of executors provided, which are `InteractiveExecutor`, `InstructExecutor` and `StatelessExecutor`.

In a word, `InteractiveExecutor` is suitable for getting answer of your questions from LLM continuously. `InstructExecutor` let LLM execute your instructions, such as "continue writing". `StatelessExecutor` is best for one-time job because the previous inference has no impact on the current inference.

Interactive mode & Instruct mode

Both of them are taking "completing the prompt" as the goal to generate the response. For example, if you input `Long long ago, there was a fox who wanted to make friend with humen. One day`, then the LLM will continue to write the story.

Under interactive mode, you serve a role of user and the LLM serves the role of assistant. Then it will help you with your question or request.

Under instruct mode, you give LLM some instructions and it follows.

Though the behaviors of them sounds similar, it could introduce many differences depending on your prompt. For example, "chat-with-bob" has good performance under interactive mode and `alpaca` does well with instruct mode.

```
// chat-with-bob

Transcript of a dialog, where the User interacts with an Assistant named Bob. Bob is helpful, kind, honest, good at wri

User: Hello, Bob.
Bob: Hello. How may I help you today?
User: Please tell me the largest city in Europe.
Bob: Sure. The largest city in Europe is Moscow, the capital of Russia.
User:
```

```
// alpaca

Below is an instruction that describes a task. Write a response that appropriately completes the request.
```

Therefore, please modify the prompt correspondingly when switching from one mode to the other.

Stateful mode and Stateless mode.

Despite the differences between interactive mode and instruct mode, both of them are stateful mode. That is, your previous question/instruction will impact on the current response from LLM. On the contrary, the stateless executor does not have such a "memory". No matter how many times you talk to it, it will only concentrate on what you say in this time.

Since the stateless executor has no memory of conversations before, you need to input your question with the whole prompt into it to get the better answer.

For example, if you feed `Q: Who is Trump? A:` to the stateless executor, it may give the following answer with the antiprompt `Q:` .

```
Donald J. Trump, born June 14, 1946, is an American businessman, television personality, politician and the 45th Pres

## Presentación previa

* Defensor del título:  Daniil Medvédev
```

It seems that things went well at first. However, after answering the question itself, LLM began to talk about some other things until the answer reached the token count limit. The reason of this strange behavior is the anti-prompt cannot be match. With the input, LLM cannot decide whether to append a string "A: " at the end of the response.

As an improvement, let's take the following text as the input:

```
Q: What is the capital of the USA? A: Washingtong. Q: What is the sum of 1 and 2? A: 3. Q: Who is Trump? A:
```

Then, I got the following answer with the anti-prompt `Q:` .

```
45th president of the United States.
```

At this time, by repeating the same mode of `Q: xxx? A: xxx.` , LLM outputs the anti-prompt we want to help to decide where to stop the generation.