retell-custom-llm-python-demo

This is a sample demo repo to show how to have your own LLM plugged into Retell.

This repo currently uses `OpenAI` endpoint, and is not as stable and fast as `Azure OpenAI` endpoint. So expect a more varying delay. Feel free to contribute to make this demo more realistic.

Steps to run in localhost

1. First install dependencies

```
pip3 install -r requirements.txt
```

2. Fill out the API keys in `.env`

3. In another bash, use ngrok to expose this port to public network

```
ngrok http 8080
```

4. Start the websocket server

```
uvicorn server:app --reload --port=8080
```

You should see a fowarding address like `https://dc14-2601-645-c57f-8670-9986-5662-2c9a-adbd.ngrok-free.app`, and you are going to take the IP address, prepend it with wss, postpend with `llm-websocket` path and use that in the dashboard to create a new `agent`. Now the `agent` you created should connect with your localhost.

The custom LLM URL would look like `wss://dc14-2601-645-c57f-8670-9986-5662-2c9a-adbd.ngrok-free.app/llm-websocket`

Optional: Phone Call Features via Twilio

The `twilio_server.py` contains helper functions you could utilize to create phone numbers, tie agent to a number, make a phone call with an agent, etc. Here we assume you already created agent from last step, and have `agent id` ready.

To ues these features, follow these steps:

1. Make sure twilio_client is initialized and `/twilio-voice-webhook/(agent_id_path)` is in `server.py` file to set up Twilio voice webhook. What this does is that every time a number of yours in Twilio get called, it would call this webhook which internally calls the `register-call` API and sends the correct audio websocket address back to Twilio, so it can connects with Retell to start the call.

2. Put your ngrok ip address into `.env`, it would be something like `https://dc14-2601-645-c57f-8670-9986-5662-2c9a-adbd.ngrok-free.app`.

3. (optional) Call `create_phone_number` to get a new number and associate with an agent id. This phone number now can handle inbound calls as long as this server is running.

4. (optional) Call `register_phone_agent` to register your Twilio number and associate with an agent id. This phone number now can handle inbound calls as long as this server is running.

5. (optional) Call `delete_phone_number` to release a number from your Twilio pool.

6. (optional) Call `transfer_call` to transfer this on-going call to a destination number.

7. (optional) Call `end_call` to end this on-going call.

8. Call `create_phone_call` to start a call with caller & callee number, and your agent Id. This call would use the agent id supplied, and ignore the agent id you set up in step 3 or 4. It automatically hang up if machine/voicemail/IVR is detected. To turn it off, remove "machineDetection, asyncAmd" params.

Run in prod

To run in prod, you probably want to customize your LLM solution, host the code in a cloud, and use that IP to create agent.