Nemo Guardrails

The Shield Against LLM Pitfalls

Colang

Colang is a Modelling language enabling Guardrails for a conversational system.

Why New Language

- Creating guardrails for conversational systems requires some form of understanding of how the dialogue between the user and the bot unfolds. Existing dialog management techniques such us flow charts, state machines, frame-based systems, etc. are not well suited for modeling highly flexible conversational flows like the ones we expect when interacting with an LLM-based system like ChatGPT.

- However, since learning a new language is not an easy task, Colang was designed as a mix of natural language and python. If you are familiar with python, you should feel confident using Colang after seeing a few examples.

Concept behind Language

- Canonical Form
- User Utterances
- Flows
- Variables
- Actions
- Rails

Colang Concepts

Canonical form

Intent of a user/bot utterance

```
define user express greeting
  "hello"
  "hi"

define user request help
  "I need help with something."
  "I need your help."
```

Utterances

The raw text coming from the user or the bot

```
define user express greeting
  "hello"
  "hi"

define user request help
  "I need help with something."
  "I need your help."
```

Flow

A sequence of messages and events, potentially with additional branching logic

```
define flow greeting
    user express greeting
    bot ask name
    bot express greeting
```

Colang Concepts

Rails

Specific ways of controlling the behavior of a conversational system.
Eg. Not talk about politics , respond in a specific way to certain user requests.

Event

Something that has happened and is relevant to the conversation
eg. User clicked something, user made a gesture etc

```
define user express greeting
  "hello"
  "hi"

define user request help
  "I need help with something."
  "I need your help."

define bot express greeting
  "Hello there!"

define bot ask welfare
  "How are you feeling today?"

define bot ask name
    "What is your name?"

define flow greeting
    user express greeting
    bot express greeting
    bot ask welfare
```

Colang Concepts

Variables

Colang allows us to use logical flows(if-else) and insert variables into our rails. There are several ways to add variables in the flow

References to context variables always start with a $ sign e.g. $name. All variables are global and accessible in all flows

Context variables are dynamically typed, and they can be: booleans, integers, floats and strings. Variables can also hold complex types such as lists and dictionaries, but they can't be initialized directly to this type of values i.e. the value would come from the return value of an action.

```
define bot ask name
    "What is your name?"

define flow greeting
    user express greeting
    if $name
        bot express greeting
    bot ask welfare
```

Colang Concepts

Actions

A custom code that the bot can invoke from the flow. It is used for multiple reasons. Eg. Third party connectivity, post processing the input or bot response

```
define flow greeting
    user express greeting
    if $name
        bot express greeting
        $answer = execute custom_action()
        bot $answer
```