NC STATE UNIVERSITY

WSPR

Automated Interaction With Smart Virtual Assistants Voice Apps Using Large Language Models

BACKGROUND AND INTRODUCTION

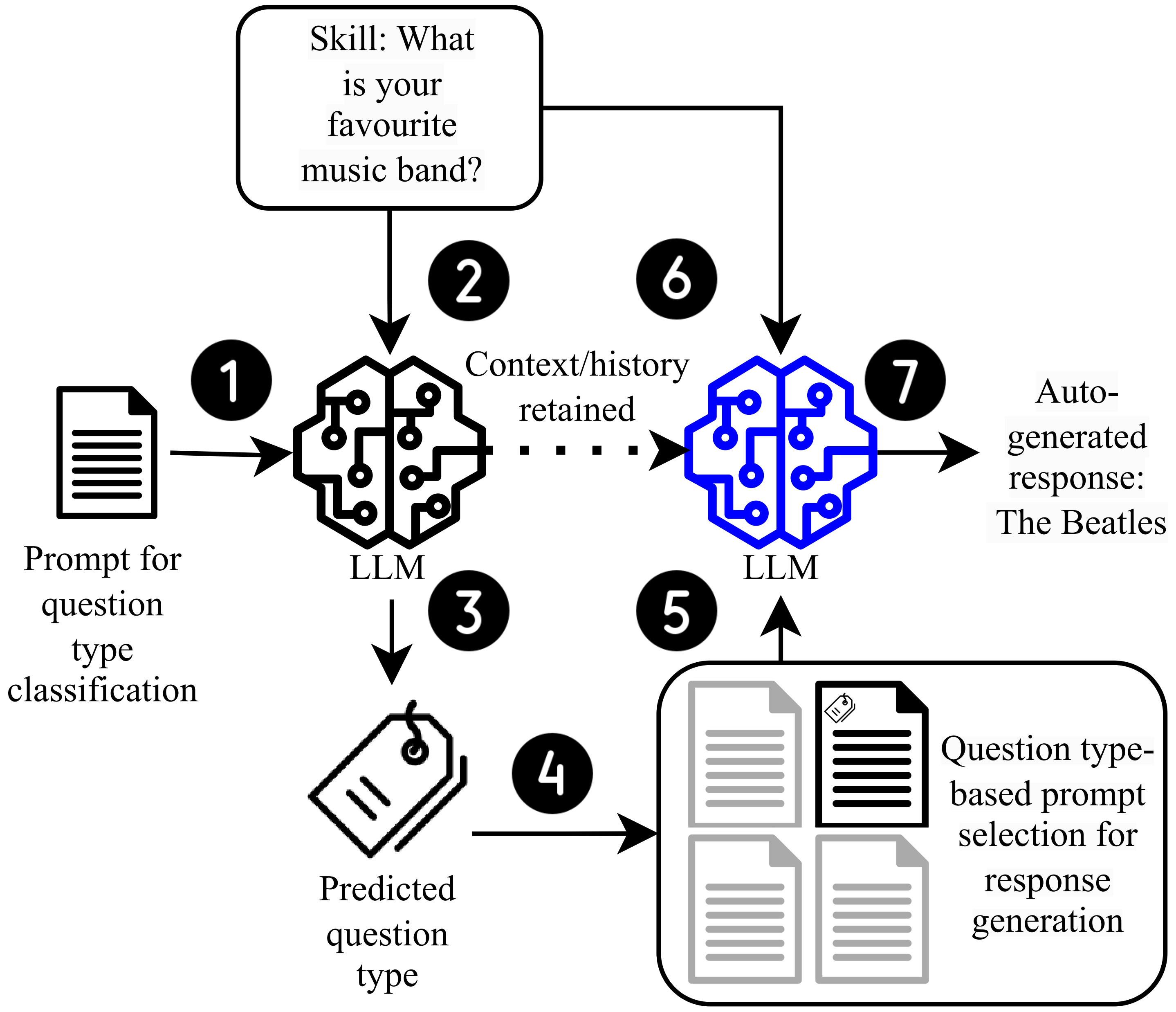
- Voice assistants support third-party apps that can contain **policy-violating content** such as hate speech, unsolicited ads, etc.
- Major VA platforms have **certification requirements** that voice apps need to comply with.
- The content of a voice app can be **dynamic** and controlled by the developer even after the voice app is published.
- **Automated interaction** with voice apps can help evaluate the content against the policy in a scalable manner.
- We aim to improve the capability of automated interaction systems to converse with voice apps to explore all content within a voice app.

METHODOLOGY

- We leverage **Large Language Models** (LLMs) to carry out conversation with a voice app.
- The LLM gets instructions via **prompt engineering** to carry out conversations with VA.
- The LLM is deployed in two phases: *1)* ***Question classification*** *phase* and *2)* ***Response generation*** *phase.*
- The response is considered correct only if the voice app proceeds according to the execution flow.
- The LLM is used in conjunction with a **driver program** that ensures all possible execution paths are explored.
- Figure 3 shows LLM's capability to generate correct responses for **non-trivial questions** via an example.

RESULTS

- We found that **LLMs can be effectively used** via prompt engineering to mimic human interaction with voice apps.
- For evaluation, we **conducted an experiment** to generate responses for Alexa skill outputs through a prompt-engineered LLM
- Our approach generated correct responses for **83.75%** of the skill outputs.
- The LLM unlike prior methods, **does not rely on fixed and finite responses** for open-ended questions.



Skill: What is your favourite music band?
1
2
3
4
5
6
7
Prompt for question type classification
LLM
Context/history retained
LLM
Auto-generated response: The Beatles
Predicted question type
Question type-based prompt selection for response generation

Figure 1: System overview

System overview description:

1) LLM is given a prompt to classify question type
2) The question is then provided to LLM
3) The LLM classifies the question type
4) The type label is used to select a prompt for response generation.
5) The selected prompt is provided to the same LLM instance to generate a response.
6) The question is provided to the LLM with instruction to generate response.
7) The LLM outputs the generated response

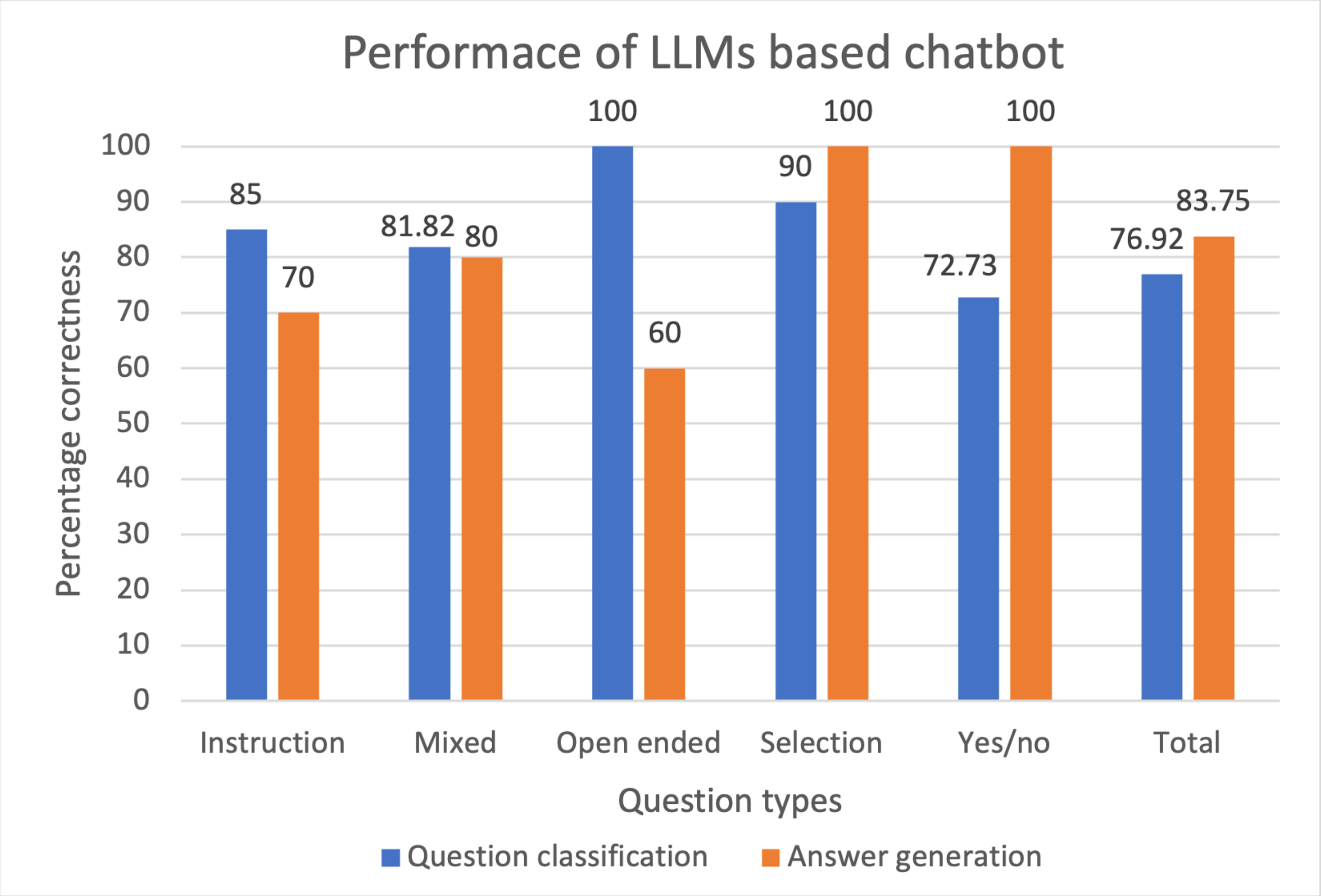
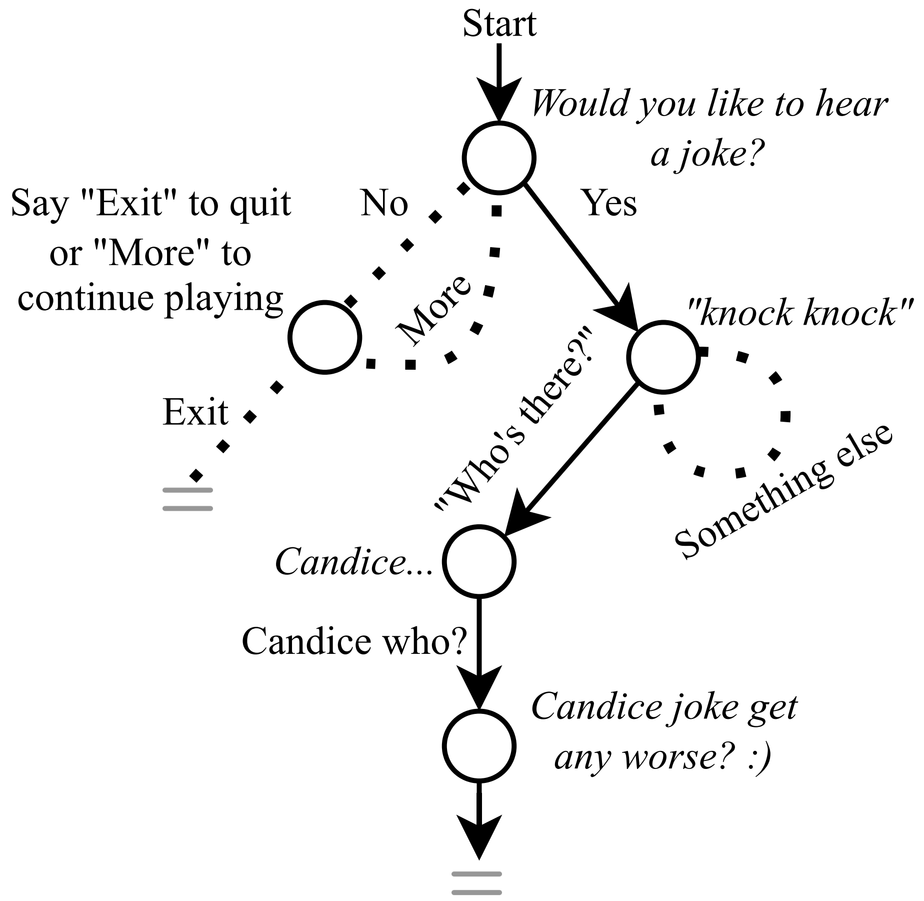


Figure 2: Response generation performance for Alexa skills



Start
Would you like to hear a joke?
No
Yes
Say "Exit" to quit or "More" to continue playing
More
Exit
"knock knock"
"Who's there?"
Something else
Candice...
Candice who?
Candice joke get any worse? :)

Figure 3: An example of a voice app execution path traversal

Example description: The skill expects the user to answer smartly to the phrase "knock knock". The LLM identifies it as an open-ended question and answers correctly "Who's there?" due to its training on large corpora of text.