Human-Machine Dialogue
Project Report

Pietro Bologna (248715)

University of Trento

pietro.bologna@studenti.unitn.it

1 Introduction

This report presents **LlamAle**, a Human–Machine Dialogue system created to assist, instruct, and inspire users in their exploration of the vast world of beer. The idea behind LlamAle is to be a friendly and knowledgeable "digital master brewer", offering users personalized beer recommendations, detailed descriptions of specific beers, and highest-rated lists.

Beer enthusiasts often face two main challenges: finding beers that match their personal taste preferences and obtaining accurate information about new styles, breweries, or ratings. LlamAle addresses both by using natural language understanding to interpret the user's questions, filling in incomplete information with targeted follow-up questions, and generating context-dependent responses. The system also supports interactive features such as keeping track of user ratings and dealing with out-of-context inquiries.

The classic exchange types for LlamAle typically include requests such as:

- "Can you recommend a Stout beer with medium alcohol content and a score above 4.0?"
- "Tell me more about Heady Topper."
- "I tried Long Trail Ale and would give it a solid 4.5. It's smooth and pleasant."

In each case, the system searches its curated beer dataset[1] for the most relevant matches, applies filtering and ranking strategies, and presents the results in a conversational format. It can also record user-provided ratings and comments, enabling future personalization.

LlamAle is designed for anyone with an interest in beer. With personalized guidance, structured search, and a friendly conversational interface, it acts as a virtual beer sommelier available anytime, anywhere.

2 Conversation Design

2.1 Types of Dialogues

LlamAle contains three main categories of dialogues, each reflecting a different way users interact with the system:

- **Recommendation and search** - The user can request beer recommendations by searching based on style, alcohol content (ABV), bitterness (IBU), or rating. This also includes looking for top-rated beers or listing beers by a specific brewery. The system ranks and organizes the results from its database before presenting them in chat format.
- **Information retrieval** – The user can ask for a specific beer to retrieve information such as style, brewery, ABV, IBU, and a short description. If necessary, the system asks follow-up questions to specify the search before returning the result.
- **Feedback and control** – The user can rate beers and provide comments, which are stored for later retrieval. The system also handles out-of-context questions gracefully and manages conversation termination when requested.

2.2 Interaction Properties

To enable these types of conversations, LlamAle employs a variety of strategies that govern how interactions are initiated, maintained, and recovered. These strategies ensure that conversations stay natural, easy to use, and robust in the face of errors or incomplete inputs.

2.2.1 Initiatives

LlamAle adopts a mixed-initiative strategy to manage the interaction flow. The user may take the initiative by providing full requests, e.g., "Recommend a stout with medium ABV and high rating." However, if the input is partial or under-specified, the system becomes active in guiding the dialogue by posing exact follow-up questions, e.g., "Do you have an ideal bitterness level?".

2.2.2 Engagement

The system engages users with a friendly tone, varied conversational markers, and in a friendly manner. It remembers relevant information from the current session to prevent repetition and ensure the interaction is smoother.

2.2.3 Acknowledgement

The system employs acknowledgement to signal to the user that his or her input has been appropriately received and understood. These cues are low-weight feedback markers, comforting the user without interrupting the conversation flow. This strategy both establishes user faith in the system's understanding and maintains the dialogue concise and on track.

2.2.4 Confirmation

Once the required slots have been filled, the system proceeds to search the beer dataset for the best possible match. By filtering results according to the extracted attributes, it ensures that the final recommendation or information provided is consistent with the user's preferences and intent.

2.2.5 Fallback answers

When the system is unable to parse the user's input or no match is returned from the dataset, it resorts to fallback strategies that keep the interaction open and user-friendly. When the user asks a non-beer question, the system moves out of the way by acknowledging its limitations while suggesting an appropriate alternative. For example, if asked "What's the capital of France? ", it can respond with "That seems a bit outside the scope of what I can help with. I'm here to chat about beer styles, recommendations, breweries, and ratings. Let me know how you'd like to explore the world of beer!". These mechanisms make the conversation meaningful even in the event of errors or out-of-scope inputs so that dead ends do not occur and user engagement is ensured.

2.2.6 User profile

LlamAle is designed to handle under-informative as well as over-informative users. When a user provides incomplete information, the system proactively seeks missing information before generating responses, ensuring that responses are correct and tailored. In case users provide too many details, or include multiple intents in one question, the system formats the input to a representational form. This allows it to generate a brief, exact answer that covers every aspect of the request, and not obscure clarity and be repetitive.

2.2.7 Conversational indicators

The system uses indicators like "Great!", "Well...", or "Alright!" to have the conversation sound natural and not robotic. They vary depending on whether it is confirming, providing information, or following up.

2.2.8 Error recovery

When the system fails to retrieve a matching response from the beer dataset, it proactively suggests a solution to the user rather than ending the interaction. In the case of an unsuccessful beer search, LlamAle summarizes the details provided, informs the user that no matching beer was found, and then asks whether the information is correct or needs to be adjusted. This strategy prevents dead ends, encourages clarification, and helps guide the conversation back on track.

3 Conversation Model

LlamAle is grounded in a four-stage pipeline that transforms raw user input into meaningful, structured conversation. The system is constructed on top of modular building blocks — Pre-NLU, NLU, DM, and NLG — that are targeted at a particular stage of the interaction. To develop all four components, LlamAle [2] relies on the Llama language model, executed locally through Ollama [3]. Running the model locally ensures fast response times (in case of good GPUs) and offers flexibility for iterative development and testing.

- **Pre-NLU** – preprocesses the input by associating it with one or more candidate intents and segmenting multi-intent utterances.
- **Natural Language Understanding (NLU)** – identifies intent and extracts slots from all intents, producing structured representations of the user request.
- **Dialogue Manager (DM)** – it keeps the state of the dialogue, ensuring all slots required are filled and determining the optimal next action. If the required data are available, the DM also interacts with the beer dataset to acquire related results so that the system can deliver correct and contextually appropriate answers.
- **Natural Language Generator (NLG)** – converts the DM's actions into natural-language responses, enriched with conversational markers to keep the interaction natural and interesting.

They collaborate together to make sure user requests are accurately interpreted, processed, and responded.

3.1 Prompts structure

The prompts in LlamAle have a uniform template structure throughout all the components. All of the prompts start with a context section that identifies the function of the component (e.g., NLU, DM, NLG) and its location in the dialogue pipeline. This is followed by a series of rules and limitations stating how the model should behave, such as the output format (typically strict JSON for NLU and DM) or response style guidelines (for NLG). Finally, the prompts are succeeded by examples of the appropriate input–output mapping, serving as a pointer so that the model can generalize to unseen cases.

3.2 Natural Language Understanding

The Natural Language Understanding (NLU) module is responsible for transforming the Pre-NLU result into structured semantic representations. It acts to identify the slot values required for facilitating the system to act and making the user requests machine-readable.

- Input – The NLU receives as input a list of intent-labeled segments from the Pre-NLU. Each segment specifies a candidate intent (e.g., get_beer_recommendation, rate_beer) together with the portion of the user's utterance associated with it.
- Processing – For every intent-tagged segment, the NLU employs an intent-specific prompt that tells the model to extract slot values relevant. The slots could be beer attributes such as style, alcohol by volume (ABV), and bitterness unit (IBU), identifiers such as brewery or beer name and user ratings such as ratings or comments. The NLU also processes special intents such as out_of_context (not beer-related questions) and terminate_system. When the slot is not explicitly mentioned, the module assigns it a null value.
- Output – The component produces a stream of JSON objects, each containing the recognized intent and a dictionary of slot-value pairs (with null for missing slots). Table 3 summarizes all possible intentions and slots.

To become more natural, the NLU has a history mechanism. This allows the system to keep slot values between turns, so that the user is not asked to repeat already given information in the same session.

3.3 Dialogue Manager

The Dialogue Manager (DM) is the central decision-making module of LlamAle. It maintains the conversation state, determines whether the user's request is sufficiently specified, and chooses the most appropriate next action. By coordinating the flow between comprehension and response, the DM ensures the interaction will be coherent and goal-directed.

- Input – The DM receives as input a list of JSON objects produced by the NLU. Each object contains an intent and its slots (some perhaps unfilled).
- Processing – The DM examines the state of each intent and decides what to do:
  1. If required slots are unfilled, it outputs a request_info action to prompt the user for the missing information.
  2. If there is sufficient information, it outputs a confirmation action and the DM interacts with the beer dataset in order to obtain the right results.
  3. If confirmation fails, the DM calls the check info action so that the request becomes clear.

  It also handles special cases such as terminate_system for the graceful ending of the dialogue and out_of_context questions.
- Output – The DM produces a list of actions, where each action is a JSON object containing: the selected "action" (e.g., request_info, confirmation, etc.), the "parameter" (e.g., the intent or slot to which the action is being applied), the "data", which contains the confirmation summary.

The DM also includes a state mechanism that keeps track of slot values across turns until the intent is complete. Once an action is successfully executed (e.g., after confirmation), the intent is removed from the state so as not to repeat. This allows continuity without posing redundant questions.

In general, the DM acts as the dialogue controller, deciding when to ask for more information, when to confirm, and when to return results. This balance of proactive slot management, confirmation, and fallback handling makes it a good basis for LlamAle's conversation flow.

3.3.1 Dataset Manager

The Dataset Manager is responsible for querying and filtering the beer dataset once the Dialogue Manager has gathered enough slot information. Its workflow is straightforward: it receives structured query parameters (e.g., style, ABV, IBU, rating), applies filtering functions, and returns a ranked list of results.

The implementation is organized around helper functions such as filtering function (e.g. filter_by_style, filter_by_abv, etc.), each of which progressively narrows down the candidate set. After filtering, the remaining beers are sorted and packaged into structured objects that can be passed to the NLG for natural-language presentation.

In short, the Dataset Manager acts as the backend engine of LlamAle, translating user preferences into precise dataset queries through modular filtering steps.

3.4 Natural Language Generator

The Natural Language Generator (NLG) module is the final piece of the pipeline. Its role is to convert the action of the Dialogue Manager in structured form to natural, conversational language that is clear, concise, and user-friendly.

- Input – The NLG receives the "action" selected by the DM (request_info, confirmation, etc.) as well as the associated "data" (slot names, confirmation strings, or database records).
- Processing – The NLG draws on action-

specific prompts to condition its answers. Each prompt specifies the style and scope of the answer:

1. For request_info, it builds a specific follow-up question to obtain missing information.
2. For confirmation, it uses the beer details contained in "data" (previously retrieved by the Dataset Manager) and produces a concise summary of those attributes.

- Output – The output is a natural-language statement consistent with the system's conversation style. For instance: "Alright! How bitter would you like the beer to be? Low, medium, or high?"

In order to maintain the conversation-like character and keep the conversation informal, the NLG employs short acknowledgement marks such as "Okay!", "Sure!", or "Great!". They facilitate continued interaction and allow the system to inform the user that it is processing the request.

In general, the NLG ensures that the internal workings of the system are made explicit in an accessible, conversational form, bridging between the formal DM outputs and the natural user expectations.

4 Evaluation

To evaluate the performance of LlamAle, two dedicated datasets were created: one for NLU evaluation and one for DM evaluation. Both datasets were initially generated with the assistance of ChatGPT to ensure diversity of examples. They were then carefully refined and corrected manually to guarantee accuracy and reliability.

- NLU eval dataset: contains annotated user utterances paired with the expected intent and slot values. The examples cover both simple and multi-intent inputs, as well as under-informative and over-informative user queries.
- DM dataset: contains sequences of NLU outputs and the corresponding expected system actions. This includes follow-up questions, confirmations, as well as cases where the dialogue requires checking slot completeness before accessing the database.

The datasets were deliberately balanced to include all system intents.

4.1 Intrinsic Evaluation

I have conducted intrinsic evaluations of the NLU and DM components. The goal was to measure how well the system identifies intents, extracts slots, and selects correct dialogue actions compared to the gold-standard annotations.

- NLU Evaluation: accuracy was measured at both the intent level and the slot level. Precision, recall, and F1-scores were also computed for each intent.
- DM Evaluation: evaluated on action accuracy and parameter accuracy, with precision, recall, and F1-scores for each action type.

The evaluation results are summarized in Table 1 and Table 2. For the NLU component, intent accuracy reached 95.89% and slot accuracy 95.97%, showing that the system reliably identifies user intents and extracts slot values.

The Dialogue Manager achieved an action accuracy of 95.95%, with strong precision, recall, and F1-scores across most actions. While intents like get_beer_recommendation were handled flawlessly, minor errors appeared in get_beer_info, mainly due to slot mismatches or alternative valid responses.

Overall, the results confirm that LlamAle performs robustly in both intent understanding and dialogue management.

4.2 Extrinsic evaluation

An informal extrinsic evaluation was conducted by letting a few friends interact with the system and later provide feedback. Users reported that LlamAle was easy to use, generally accurate in its answers, and maintained a natural and engaging conversational style.

At the same time, participants suggested improvements such as expanding the beer database, adding more variety in response phrasing, enabling persistent user profiles, and making better use of dialogue history for context. They also noted that response speed was sometimes unsatisfactory, mainly due to the modest GPU used for local execution.

5 Conclusion

This report has presented LlamAle, a human–machine dialogue system designed to help users explore the world of beers with natural dialogue. Guided by a modular pipeline of Pre-NLU, NLU, DM, and NLG, the system conducts user input interpretation, dialogue management, information retrieval on beers, and fluent response generation.

Evaluation showed good performance with high slot and intent accuracy. Informal testing confirmed the system is accurate but that users found response times to be slower and suggested adding more diverse responses and expanding the dataset to improve it.

LlamAle in general demonstrates how large language models can be integrated into structured dialogue pipelines to generate domain-specific assistants, in this instance acting as a digital beer sommelier.

Appendix

	Accuracy	Result
1	NLU Intent	95.89%
2	NLU Slot	95.97%
3	DM Action	95.95%

Table 1: *Overall intrinsic evaluation results for the system, showing average accuracy across NLU intent classification, NLU slot filling, and DM action prediction.*

	Intent	Precision	Recall	F1-Score
1	get_beer_recommendation	1.00	1.00	1.00
2	get_beer_info	1.00	0.79	0.88
3	list_beers_by_brewery	1.00	1.00	1.00
4	get_top_rated	1.00	1.00	1.00
5	rate_beer	1.00	0.91	0.95
6	out_of_context	1.00	1.00	1.00
7	terminate_system	1.00	1.00	1.00

Table 2: *Per-intent evaluation metrics (precision, recall, F1-score) for the NLU, showing detailed performance across all supported intents.*

	Intent	Slots
1	get_beer_recommendation	*style (str)*: The beer style requested (e.g., stout, IPA, lager). *abv (str)*: Preferred alcohol-by-volume range (e.g., low, medium, high). *ibu (str)*: Desired bitterness level in IBUs (e.g., low, medium, high). *rating (float or list of float)*: Target rating as a value or range (e.g., 4.0 or [4.0–5.0]).
2	get_beer_info	*name (str)*: Name of the beer. *brewery (str)*: Producing brewery (optional).
3	list_beers_by_brewery	*brewery (str)*: Brewery for which to list beers.
4	get_top_rated	*style (str)*: Style for which to retrieve top-rated beers.
5	rate_beer	*name (str)*: Beer being rated. *rating (float)*: User's numeric rating (e.g., 4.5). *comment (str)*: Optional free-text note.
6	out_of_context	–
7	terminate_system	–

Table 3: *Mapping of system intents to their corresponding slots, with brief slot descriptions.*

References

[1] N. Hould, “Craft beers dataset,” https://www.kaggle.com/datasets/nickhould/craft-cans?select=beers.csv, 2017.

[2] M. AI, “Llama 3: Open foundation and fine-tuned chat models,” 2024, available at https://www.llama.com/models/llama-3/.

[3] O. Inc., “Ollama: Chat and build with open models,” 2025, available at https://ollama.com/.