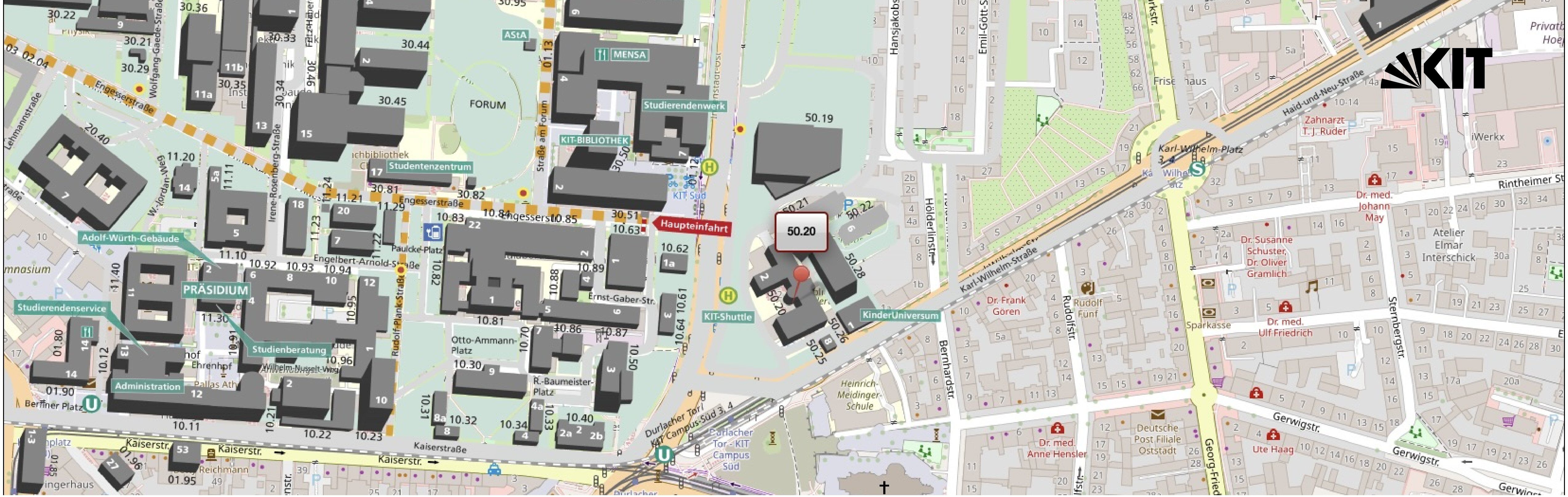


Campus Plan Bot

Practical: Natural Language Dialogue Modeling

Thomas Marwitz and Frederik Schittny | 23. July 2025

1. The Campus Plan

Current state
LLM Integration

2. System Evaluation

Evaluation Data
BERT Score
LLM-as-a-Judge

3. A First Prototype

Basic Data Flow

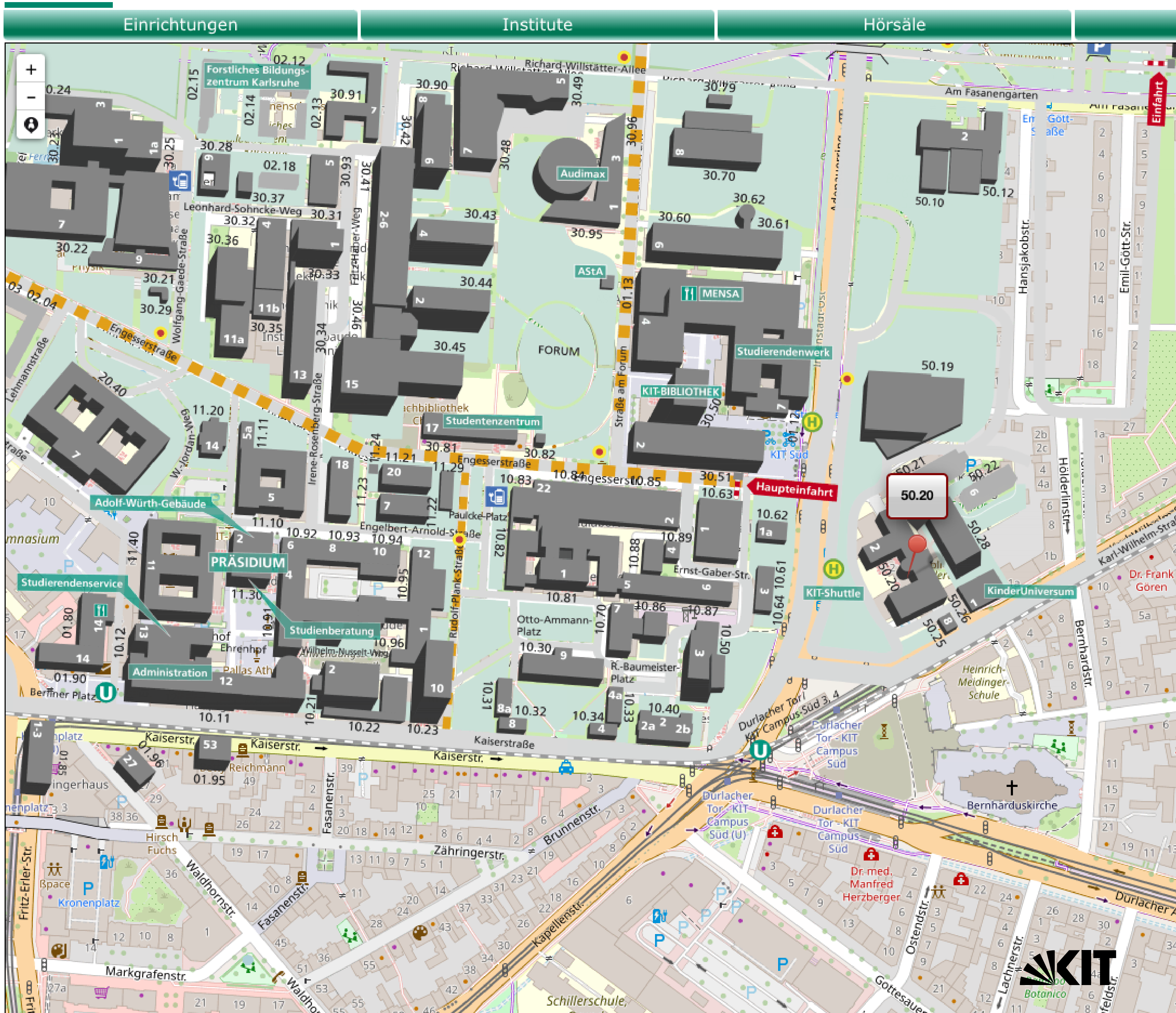
4. Data Flow Improvements

Identified Problems
Implemented Solutions
Final Evaluation

5. Demo

Old Campus Plan

- No addresses
- No interactive view
- No navigation
- No additional information



LLM Integration

- System interaction with natural language
  - ASR and textual input
- RAG-based with expanded database
  - Reverse geocoding (addresses)
  - OpenStreetmap data (e.g. opening hours, wheelchair accessibility)
- Navigation with established services
  - Navigation links for Google Maps
- Use of contextual information
  - Current time
  - (Current relative position)

Evaluation Data

1. Collect additional data
2. Data cleanup
3. Design prompt templates
   - Single-turn
   - Multi-turn
4. Slot filling
5. LLM-assisted rephrasing
6. Record audio samples
7. System evaluation strategy

BERT Score

- Precision and recall based on dense embeddings
- Meant to measure semantic similarity
- Not precise enough for our system
  - Focused too much on word similarity
  - Assigns high scores to counterfactual responses
  - Too hard to distinguish good from bad responses

An Example

Input: Ist Gebäude 210 rollstuhlgerecht?
Expected Output: Ja, das Gebäude ist rollstuhlgerecht.
Actual Output: Das Gebäude 210 ist nicht rollstuhlgerecht.

BERT Score: FScore: 0.87 (precision: 0.87; recall: 0.86)

LLM-as-a-Judge

- Use LLM to compare expected and actual response
- Flexible scoring options

Scores we use:

- Pass/Fail score
  - Basic measure for test cases
  - Easiest to evaluate improvements
- Quality score + judge explanation
  - Continuous scale from 0 to 1
  - Sensitive to quality changes not reflected in pass/fail change
  - Explanation analysis can help identify issues

Challenges:

- LLM judge capabilities
  - Small models are not powerful enough
- Alignment
  - Identifying task intention
  - Subtracting points for "bad style"
  - Ignore excuses made by system

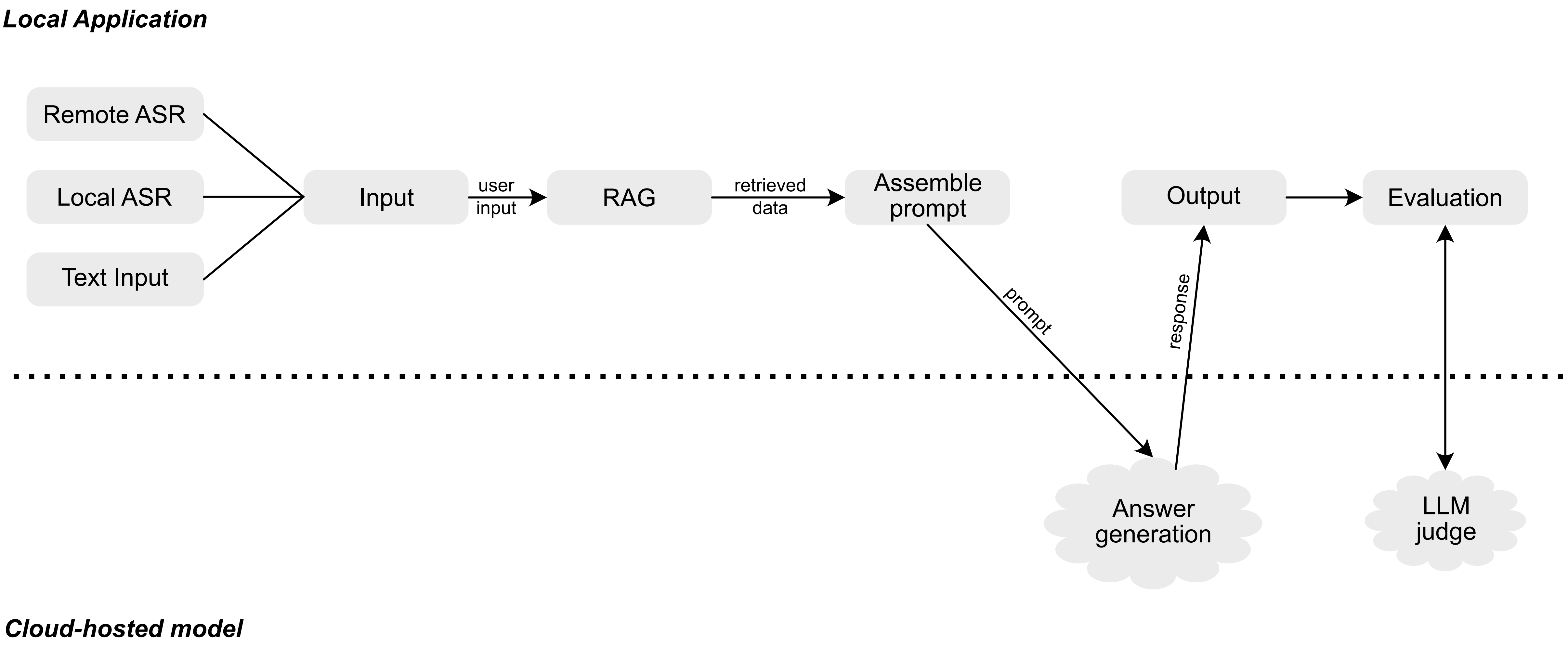
A First Prototype

- Minimum Viable Product (MVP)
  - One (basic) version of every core component
  - Command line interface
- Componentization with Python protocols
  - Easy to iterate on individual components

Core Components:

- Input
  - Options: text, local ASR, remote ASR
- Document retrieval (RAG)
  - Cosine similarity of embeddings
  - RegEx for numerical building IDs
- Prompt assembly
  - System prompt
  - User query + conversation history
  - Retrieved documents
  - Current time
- Answer generation
- Output

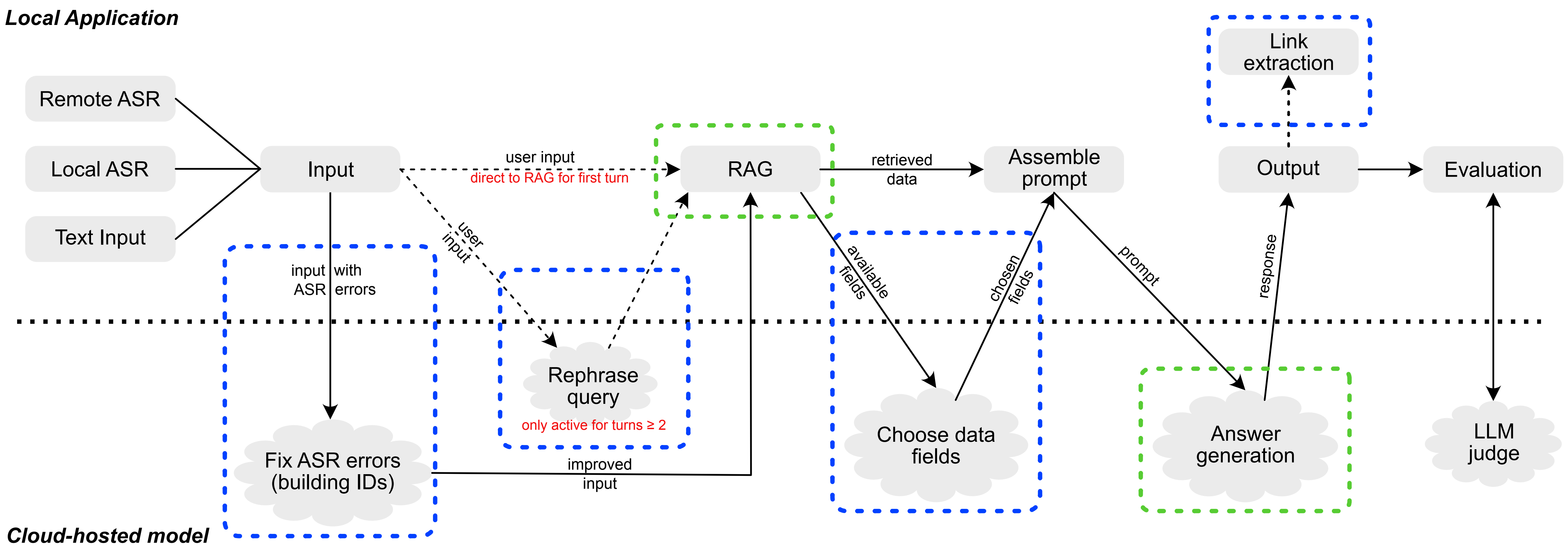
Basic Data Flow



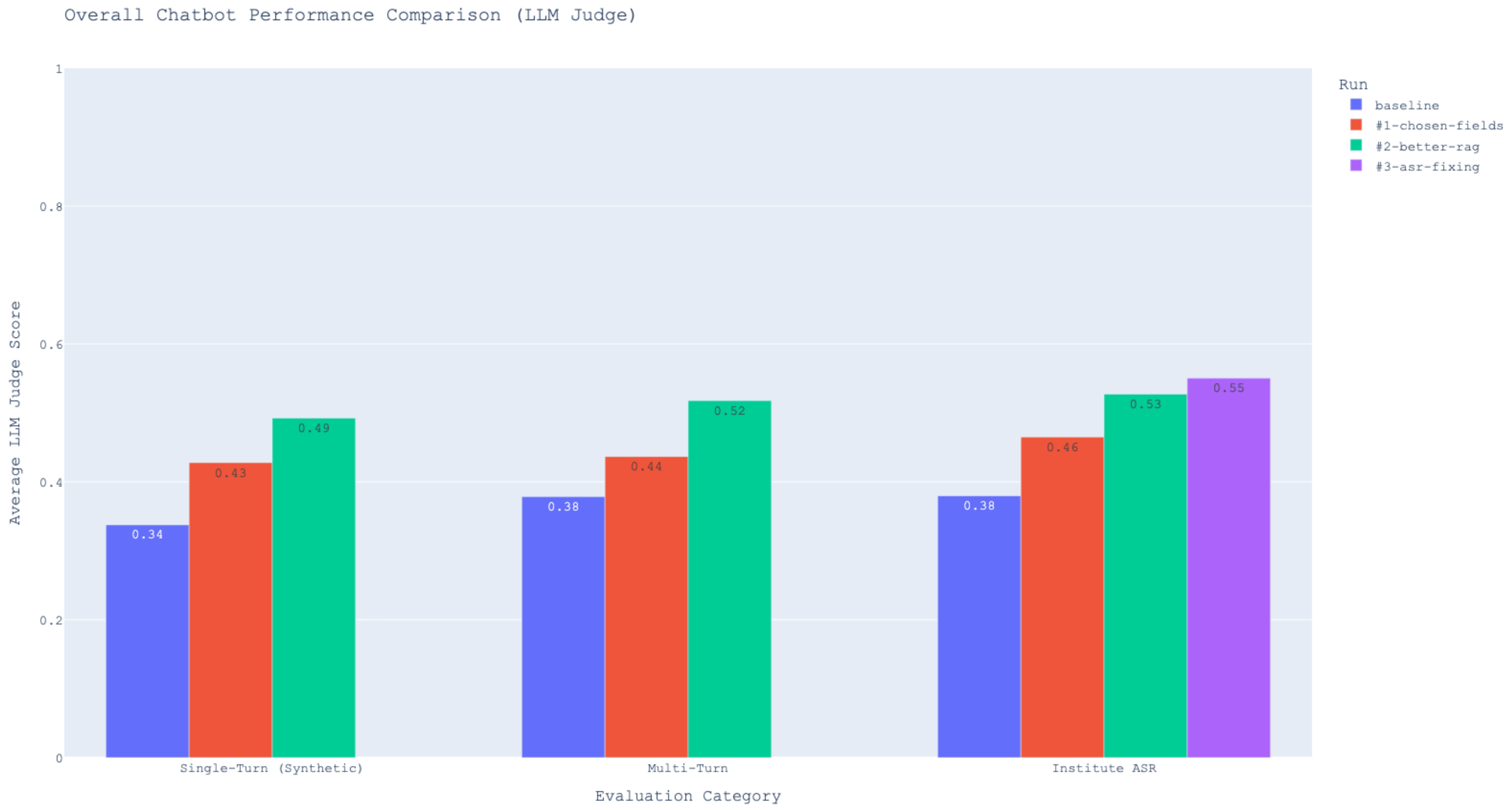
Identified Problems

- ASR errors
  - High impact on retrieval
  - Especially building IDs (e.g. "Gebäude fünfzig Punkt zwanzig")
- Missing multi-turn context
  - Some queries rely on context
  - No successful retrieval possible
  - Model has to attend to conversation history
- Inaccurate retrieval
  - Embeddings unfit for matching numerical IDs
- Too much returned information
  - Model tends to use all provided data
  - Unnecessary information in response
- Suboptimal system prompt
  - Language mismatch
  - Low structure
  - Instruction order

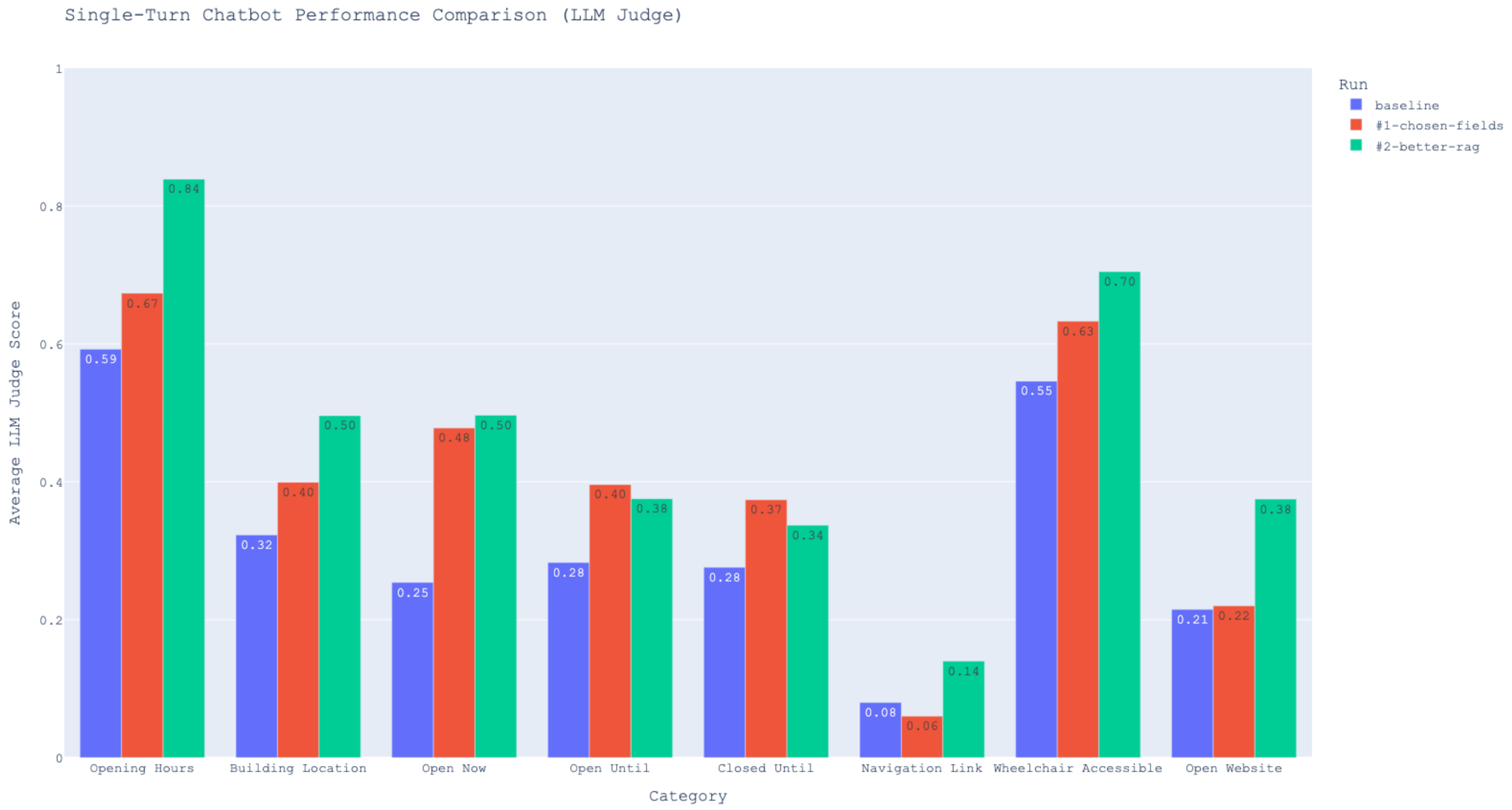
Implemented Solutions



LLM Judge Score



LLM Judge Score



Pass/Fail Score

Category	# Test Cases	Baseline	Improvements
Building Location	85	19	40
Closed Until	100	24	31
Navigation Link	50	0	0
Open Now	100	20	45
Open Until	100	21	33
Open Website	100	17	34
Opening Hours	100	53	81
Wheelchair Accessible	100	49	68

Demo Time