NBA Fan AI Agent: Natural Language Understanding System for NBA Fans

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Kevin Zhou, Jiadong Zhu

CSE 595: Natural Language Processing

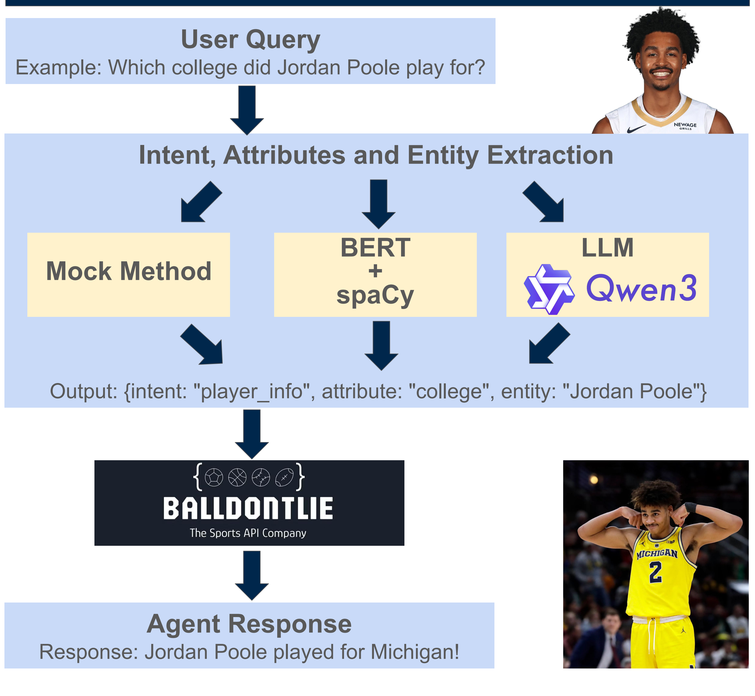
Introduction

- NBA fans often spend significant time searching online for game information and player statistics. Existing AI assistants are limited to basic queries, while newer systems often rely on slow web retrieval.
- **Our solution** is to create a domain-specific NBA Fan AI Agent that delivers fast, reliable answers through natural language understanding (NLU), combining intent classification and slot filling to map queries to structured information.

NLP Task Definitions

- The core NLP task is Natural Language Understanding (NLU), combining:
  - Intent Classification: Maps queries to categories (e.g., game schedule, player info, team info)
  - Slot Filling: Extracts structured information including attributes and entities (player/team names)

System Architecture



Dataset

- **300 annotated queries** covering NBA player and team info
- **2 intent classes**: player_info (200), team_info (100)
- **15 attributes**: 10 player info (position, height, weight, jersey_number, college, country, draft_year, draft_round, draft_number, team) + 5 team info (conference, division, city, full_name, abbreviation)
- **20 examples per attribute** with varied phrasings
- **Train/Val/Test**: 80/10/10 split. Train/Val used for BERT fine-tuning. Test used for evaluation of all NLU methods
- **Format**: JSON with <name> placeholder for all entities (player/team names)

Methods

1. **Mock Method**
   - Rule-based baseline
   - Intent detection via **regex patterns** for team/player keywords
   - Attribute extraction through **keyword-to-attribute mapping**
   - Entity extraction via hardcoded keyword lists (30 teams, 20+ players) and regex pattern matching
   - Fast inference, no training required
2. **BERT**
   - Fine-tuned **bert-base-uncased** with dual classification heads
   - Architecture: BERT encoder + intent head (2 classes) + attribute head (15 classes)
   - Training: 15 epochs, AdamW optimizer
   - Uses spaCy for entity extraction
   - **spaCy**
   - **Named Entity Recognition (NER)** using en_core_web_trf transformer model. Extracts player names (PERSON) and team names (ORG/GPE) from queries. Used by BERT
3. **LLM**
   - **Qwen3-4B-Instruct-2507-FP8** (4B params, FP8 quantized) with zero-shot in-context learning
   - **Prompts engineering** with explicit rules
   - Single-step prediction (intent, attribute, and entity)
   - No fine-tuning required; handles entity extraction directly without spaCy

Results

Method	Test Accuracy	Inference Time (ms)
Mock Method	66.67%	0.18
BERT + spaCy	73.33%	47.15
LLM (Qwen3)	96.67%	1573.16

- Mock Method is fastest but least accurate due to rule-based limitations.
- LLM achieves the highest accuracy via zero-shot prompting, but is slowest due to generation overhead.
- BERT + spaCy balances accuracy and speed with fine-tuning.
- Results show clear accuracy-speed trade-off in methods.

End-to-End System

- Integration with BALLDONTLIE API:
  - Entity Linking: Fuzzy matching + alias resolution. Maps entity names to API IDs
  - API Routing: Intent-based endpoint selection. Retrieves structured data from Ball Don't Lie API
  - Response Formatting: Converts API data to responses

Future Work

- Add *"game_info"* intent support
- Build interactive web UI for user-friendly interaction
- Enhanced error handling for edge cases
- Evaluation on natural user queries

Conclusion

Key Advantages over General LLMs (e.g., ChatGPT)

- **Speed**: Our model delivers results faster, while ChatGPT requires web search and token-by-token generation, making responses significantly slower.
- **Accuracy**: Specialized NLU achieves high accuracy on domain-specific queries with limited training data, while maintaining interpretability.
- **Efficiency**: Lightweight BERT+spaCy pipeline suitable for real-time dialogue applications, without the computational overhead of large language models.