

Homework 7

Add Natural Language Interactivity to Uniswap UI

Project Overview

This project extends the Uniswap UI from Homework 6 with natural language interactivity, allowing users to interact with Uniswap contracts through conversational commands. The application interprets natural language instructions to perform operations like depositing liquidity, redeeming tokens, and executing swaps, as well as answering queries about pool data.

1 Key Features

1.1 Natural Language Processing

- Conversion of natural language instructions to structured function calls
- Support for all core Uniswap operations through conversational commands
- Data analysis capabilities for Uniswap pool metrics and events

1.2 LLM Integration

- Dual integration with OpenAI's models and open-source LLMs
- Side-by-side comparison of performance between models
- Custom API endpoint configuration for using your own LLM

1.3 Transaction Flow

1. NL instruction parsing to structured representation
2. Conversion to Ethereum transaction parameters
3. Transaction signing with MetaMask integration
4. Submission to Ethereum network via JSON-RPC

1.4 Evaluation Framework

- 10 standard test cases with varying complexity

- 10 challenging test cases to demonstrate system limitations
- Interactive test case execution with result comparison
- Capability to add custom test cases through the UI

2 Contract Deployment

All smart contracts have been deployed and verified on the Sepolia testnet:

Contract	Address
TokenA	0xB667f5e8171468F12F47eF7E03C76F5594F33248
TokenB	0xC215279133800AFca77eF50C6651db656831138e
TokenATokenBPool	0x117F180f4bB2235c075b4a3FB5Dc5ff72d74A739

3 Development Setup

To run the application locally:

```
1 # Start local blockchain
2 yarn chain
3
4 # Deploy contracts
5 yarn deploy --reset
6
7 # Start frontend application
8 yarn start
```

Listing 1: Setup Commands

4 Open Source LLM Integration

The application allows users to select or configure their preferred LLM:

- Default OpenAI integration
- Support for locally hosted open-source models
- Custom endpoint configuration via UI settings
- Performance comparison metrics across different models

5 Public Deployment

The application is publicly accessible at:

<https://tinyuniswap-llm.vercel.app/>

6 Source Code

All source code is available on GitHub:

github.com/chongchen1999/TinyUniswap-LLM.git

7 Future Improvements

- Enhanced error handling for complex or ambiguous natural language instructions
- Support for more advanced pool analysis queries
- Integration with additional open-source LLMs
- Improved performance optimization for on-chain data analysis