

Natural Language Interface for Classic Text Adventures

Making Zork Understand Human Language

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with a
front door.
There is a

CSPB-4830, Spring 2025 | Josiah Panak

The Problem

Rigid Command Syntax in Text Adventures

Traditional Requirements:

- Exact command syntax
- Memorized vocabulary
- Specific word order

Opening the small mailbox reveals a leaflet.

```
>what does the leaflet say?  
I don't know the word "does".
```

I'll be the one who's going to bring peace and order to this realm. I want to be remembered for it.

I just want to help people. If I can make this area safer, I'm going to do it.

Player Frustration: Fighting the parser instead of solving puzzles

Project Goals

Primary Objective: Build an NLP system for natural, conversational gameplay

Research Questions

- Can we map varied natural language to fixed commands?
- How to handle ambiguity without game state?
- Optimal balance between flexibility and accuracy?

Success Metrics

- >90% command accuracy
- <100ms response time
- Seamless experience / implementation

Dataset & Testing

Custom Test Set

- 64+ command variations
- 15 action types covered
- Natural conversational phrases + Zork intended syntax

Example Test Cases

- "I'd like to pick up the lamp" → take lamp
- "Look inside the mailbox?" → open mailbox
- "How am I doing" → diagnose
- "What am I carrying?" → i

Technical Approach

Hybrid Parsing Strategy

Rule-Based:

- Pattern Matching
- Keyword Detection
- Synonym Dictionaries

Embedding-Based:

- Semantic Similarity (*Sentence Transformers*)
- Handles Unknown Phrases
- Flexible Adaptation

Pipeline:

Input →

Remove Fillers →

Extract Action →

Extract Target →

Generate Command

Tools & Implementation

Core Technologies

- SentenceTransformers (all-MiniLM-L6-v2)
- scikit-learn (cosine similarity)
- Python regex patterns
- pexpect (game integration)
- Frotz (Z-machine interpreter)

Key Innovation

Affordance extraction: System learns what actions work with which objects.

Experiments

Basic (first attempt)

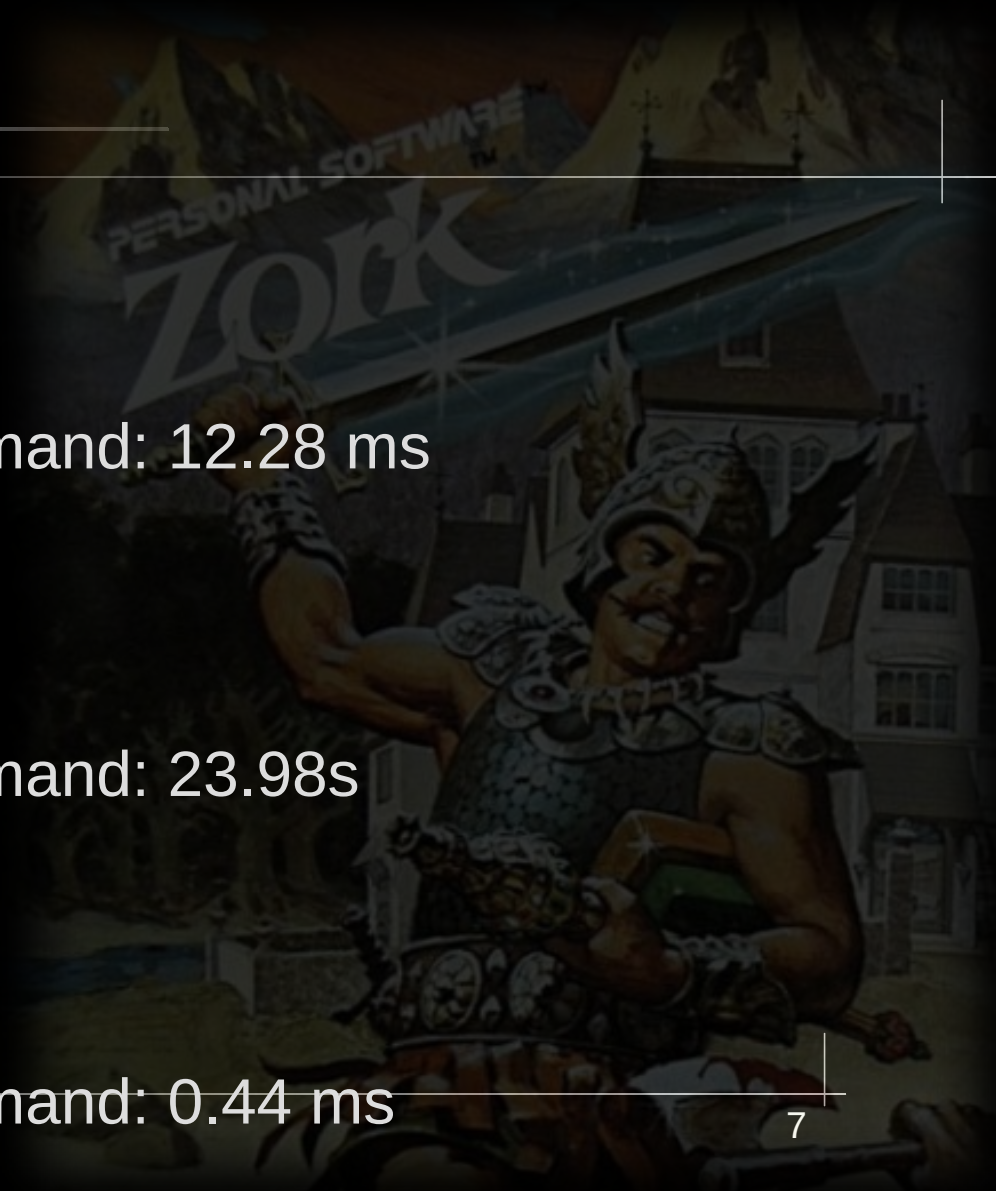
- Test Cases Passed: 40%
- Average response time per command: 12.28 ms

Context Aware:

- Test Cases Passed: 32.3%
- Average response time per command: 23.98s

Enhanced:

- Test Cases Passed: 100%
- Average response time per command: 0.44 ms



Results

Performance Achieved

- All single command test cases passed with >0.9 confidence.
- Only Exception being tests with multiple commands or longer commands where the verb and object isn't clear

```
>walk to the east  
You used the word "east" in a way that I don't understand.
```

```
> Walk to the east  
[→ e]  
The door is locked, and there is evidently no key.
```


Challenges & Lessons

The background of the slide is a dark, textured image. On the right side, there is a prominent image of a Zork game box, showing the title 'Zork' in a large, stylized font and 'PERSONAL SOFTWARE' above it. To the left of the box, there is a character wearing ornate, metallic armor and a helmet with a crest, holding a sword. The overall aesthetic is that of a classic role-playing game.

Technical Challenges

- Fully adaptive, context aware processor
- Complex commands with multiple interpretations
- Multi-step commands

Key Insights

- Stateless is limited, but surprisingly effective
- Players adapt to what works

Possible Improvements

- Lightweight state tracking (inventory, named npcs, etc.)

Conclusion

Achievement: 40-year-old Zork playable with natural language

Contributions

- Hybrid parsing approach
- <100ms working implementation
- Nearly flawless accuracy with limited guidelines
- Applied academic research to practice with success