WikiRacer

A semantic Wikipedia navigator that finds paths between any two Wikipedia articles using AI-powered embeddings.

What is WikiRacer?

WikiRacer automatically navigates from one Wikipedia page to another by analyzing links and using semantic similarity to choose the best path. It uses sentence transformer embeddings to understand the meaning of link text and find connections that are semantically related to the target page.

For example, navigating from "Potato" to "Goat" might follow a path like:

```
Potato → Chicken → Rice → Dairy → Goat
```

Features

- **Semantic Navigation**: Uses AI embeddings (sentence-transformers) to find semantically similar links
- **Vector Database**: Stores embeddings in ChromaDB for fast similarity search
- **Loop Prevention**: Tracks visited pages to avoid infinite loops
- **Live Visualization**: Optional browser-based demo mode showing real-time navigation with highlighted links
- **Path Logging**: Tracks and displays the complete path taken

Installation

1. Clone the repository
2. Install dependencies:

```
pip install -r requirements.txt
```

Usage

Run the main script:

```
python main.py
```

You'll be prompted to:

1. Choose whether to see the visual demonstration (y/n)

2. Enter the starting Wikipedia URL
3. Enter the target Wikipedia URL

Example

```
Would you like to see the visual demonstration? (y/n): n

Enter the START Wikipedia URL:
https://en.wikipedia.org/wiki/Python_(programming_language)
Enter the TARGET Wikipedia URL:
https://en.wikipedia.org/wiki/Java_(programming_language)
```

Demo Mode

When you select demo mode (`y`), a browser window opens showing:

- The Wikipedia page being viewed
- A sidebar with the navigation path
- Real-time highlighting of the next link to be clicked
- Status updates as the algorithm works

How It Works

1. **Scrape**: Extracts all Wikipedia article links from the current page
2. **Embed**: Creates semantic embeddings for each link name using `all-MiniLM-L6-v2`
3. **Search**: Finds the link most semantically similar to the target page name
4. **Navigate**: Moves to that page and repeats until the target is found or max depth is reached

The algorithm:

- Checks if the target page is directly linked (instant win)
- If not, uses cosine similarity to find the closest match
- Excludes already-visited pages to prevent loops
- Stops after 20 steps if target isn't found

Project Structure

```
wikiracer/
├── main.py            # Entry point and WikiRacer class
├── html-scrape.py     # Wikipedia page scraper
├── embeddings.py      # Embedding storage and similarity search
├── visualizer.py      # Browser visualization server
├── viewer.html        # Visualization UI
├── requirements.txt   # Python dependencies
└── README.md
```

Dependencies

- `requests` - HTTP requests for Wikipedia
- `beautifulsoup4` - HTML parsing
- `sentence-transformers` - Semantic embeddings
- `chromadb` - Vector database
- `websockets` - Real-time visualization communication

Configuration

In `main.py`, you can adjust:

- `max_depth`: Maximum steps before giving up (default: 20)

In `visualizer.py`, you can adjust:

- `http_port`: Port for the visualization server (default: 8080)
- `ws_port`: WebSocket port (default: 8765)

Limitations

- Only works with English Wikipedia (`en.wikipedia.org`)
- Semantic similarity isn't perfect - sometimes takes indirect paths
- Large pages with many links take longer to process
- Network-dependent performance

Credit

Project inspired by: https://github.com/theGreen-Coder