PROJECT WORK REPORT

by Akan Zholdygali, Yelaman Seiitkhanuly

Technology stack Backend (Spring Boot), Frontend (React+Vite)

1. Trie-based Auto-Completion Search

The project started with implementing classic prefix-based search using Trie data structures. Two variants were implemented: TrieSet for storing words and TrieMap for associating words with values. Insertion and lookup operations run in O(L), where L is the word length. Auto-completion is implemented using Depth-First Search (DFS) starting from the prefix node, collecting all valid words below it. This approach demonstrates efficient exact and prefix search but lacks semantic understanding of text.

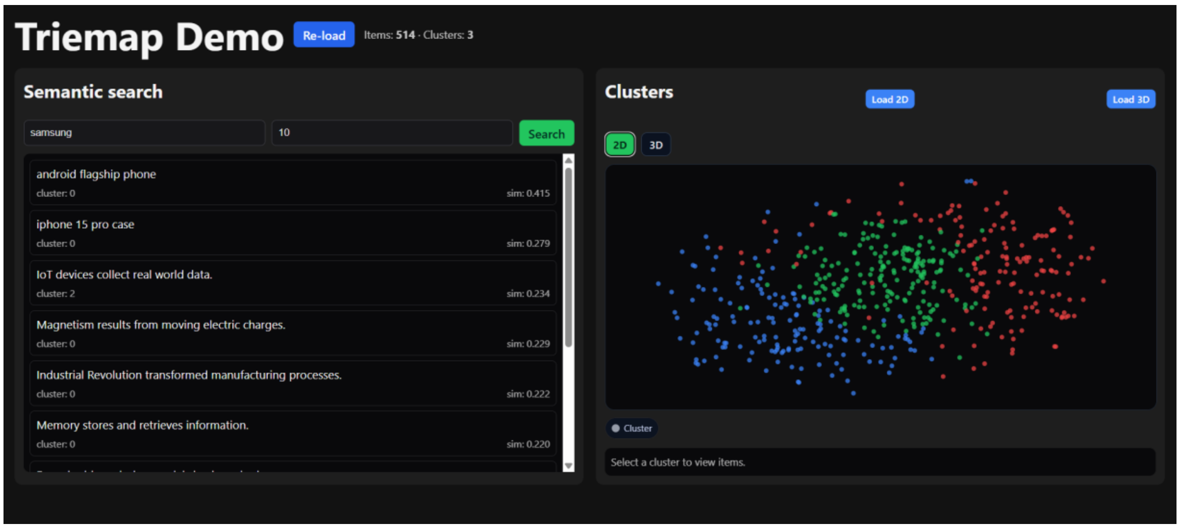
2. Semantic Search from Scratch

The second stage introduced semantic search without external libraries. Texts were vectorized using simple character-based vectors and Bag-of-Words representations. K-Means clustering was implemented manually to group similar texts into a fixed number of clusters. For search, cosine similarity was applied between query and documents, but only inside the closest clusters, reducing search space and improving efficiency. This stage demonstrated how clustering and vector similarity can approximate semantic search.

3. Extended Semantic Search with NLP Libraries

In the final stage, the project was extended using modern NLP libraries. Pre-trained embeddings were integrated using SMILE's GloVe model and DeepLearning4J's Word2Vec. These embeddings capture semantic meaning beyond exact word matches. Clustering and dimensionality reduction were upgraded using SMILE's K-Means and t-SNE, allowing 2D and 3D visualization of text clusters. This evolution transformed the project from classical data structures into a modern semantic search and visualization system.

Demo



android flagship phone
cluster: 0 sim: 0.415

iphone 15 pro case
cluster: 0 sim: 0.279

IoT devices collect real world data.
cluster: 2 sim: 0.234

Magnetism results from moving electric charges.
cluster: 0 sim: 0.229

Industrial Revolution transformed manufacturing processes.
cluster: 0 sim: 0.222

Memory stores and retrieves information.
cluster: 0 sim: 0.220