Book Recommender using NLP

Carsten Lydekin

Zealand Business Colleg

Oral Exam – AI and ML, 4th Semester

Book Recommender using NLP

Constan Lyabiding
Zashard Hustman College
Oral Exam – AI and ML, 6th Sementer

OPENING (30 seconds):

- Good morning, I'm Carsten
- Book recommender using NLP for oral exam
- 10 min presentation + questions
- Focus on concepts and implementation

Motivation and Problem



Book Recommender using NLP

b Good: Build a fully offlice contembored book recommender

► Key Requirements: No external APIs, no user tracking, semantic understanding

- ▶ Growing privacy and cost concerns with cloud-based systems
- ► Traditional systems require user profiles and data collection ▶ Goal: Build a fully offline, content-based book recommender
- **▶** Research Question:

How can a local ML model recommend books based on natural language descriptions?

Key Requirements: No external APIs, no user tracking, semantic understanding

1. Welcome to the talk!

└─Motivation and Problem

2. As you can see, this slidedeck is a work in progress.

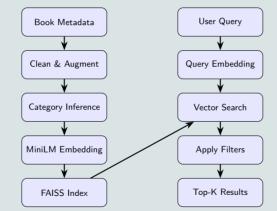
Architecture Overview



Core Components:

- ▶ Data Pipeline: Clean & augment book metadata
- ► Category Inference: Zero-shot classification + fallback rules
- ► Semantic Embedding: MiniLM sentence transformers
- ► Vector Search: FAISS similarity matching
- ► Local UI: Streamlit interface

Key Innovation: Fully local semantic search without cloud dependencies



Book Recommender using NLP

Architecture Overview

Core Components:

Data Pipeline: Clean & augment book metadata Query Embedding
Ventor Snanh
Apply Films
Top K Reachs ➤ Category Inference: Zero-shot classification + ➤ Semantic Embedding: MIAILM sentence transformers
➤ Vector Search: FAIRM similarity matching

► Local UE Streamlit interface Key Important fully local persentic search without

Dataset Exploration & Processing

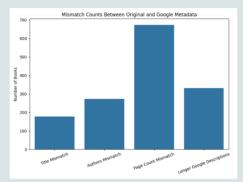


Data Quality Challenges:

- ► Started with 6,800 books from multiple sources
- ► Issues found:
 - Missing author/category information
 - ∨ Very short descriptions (< 9 words)
 </p>
 - ▶ Inconsistent categorization across sources

Data Engineering Solutions:

- ► API enrichment: OpenLibrary & Google Books
- ▶ Quality filtering: Remove inadequate descriptions
- ► Final dataset: 5,160 high-quality books



Data inconsistencies across sources required systematic cleaning and validation

Book Recommender using NLP

Dataset Exploration & Processing

Data Quality Challenges: - Ironer found

Data Engineering Solutions:

• API environment: OpenLibrary & Google Books

► Quality filtering: Remove inadequate descriptions ► First dataset: 5.160 high-quality books

Category Inference Strategy

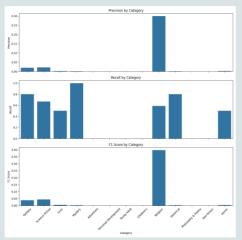


Two-Tier Classification Approach:

- ▶ Primary: Zero-shot classification with BART-MNLI
 - ▶ No training data required
 - ▶ 13 predefined book categories
 - Confidence scoring for predictions
- ► Fallback: Rule-based keyword matching
 - ▶ When confidence < threshold</p>
 - ▶ Genre-specific keyword patterns

Quality Control:

- ▶ Description length ≥ 200 chars
- ► Average confidence ≥ 0.2
- ► Maximum confidence ≥ 0.4



Results focus on high-confidence predictions rather than perfect recall across all categories

Book Recommender using NLP

Category Inference Strategy

Two-Tier Classification Approach: ► Primary: Zero-shot classification with BBRT-NO

Confidence scoring for predictions . Enthoris: Bala-based knowned marchine

► Description learth > 200 char

- ► Average confidence > 0.2
- ► Maximum confidence > 0.4





Semantic Embedding & Vector Search



Sentence Embedding with MinilM:

► Model: all-MiniLM-L6-v2

► Input format:

"Title: ... Author: ... Description: ..."

▶ Output: 384-dimensional vectors

► Advantage: Semantic similarity beyond keywords

Vector Search with FAISS:

▶ Index: 5,160 book embeddings

► Search: L2 distance (exact search)

▶ Performance: < 10ms query time

► Local: No external dependencies



End-to-end semantic search in < 200ms

Book Recommender using NLP

Semantic Embedding & Vector Search

Sentence Embedding with MiniLM:

- ➤ Input format:
 "Title ... Author ... Description: - Output: 184-dimensional vectors
- Advantage: Segretic similarity beyond knowners Verter Search with FATRS:

- ► Index: 5.160 book embeddings
- ➤ Performance: < 10ms query time as Local: No external dependencies

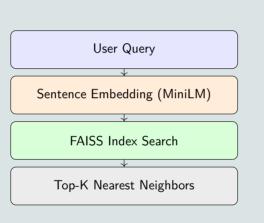


Query: "space exploration novels

Vector Similarity Search with FAISS



- ► Used Facebook AI Similarity Search (FAISS) library
- ▶ Performs **L2 distance** search in embedding space
- ► Index built with:
 - \triangleright 5160 book embeddings ($\vec{v} \in \mathbb{R}^{384}$)
 - Exact search (IndexFlatL2)
- ► At runtime:
 - User guery is embedded
 - ▶ Top-k nearest neighbors retrieved
 - Results shown in UI
- ► Fully local, fast search



Book Recommender using NLP

└─Vector Similarity Search with FAISS

User Query ____ Sentence Embedding (MiniLM) FAISS Index Search Top-K Nearest Neighbors

b. Hard Euroback Al Similaria Search (EAISS)

At nartime:
 User query is embedded
 Top-is nearest neighbors retrieved
 Results shown in UI

Fully local, fast search

Critical Analysis & Future Directions



Implementation Strengths:

- ▶ Privacy-preserving by design
- ► Semantic understanding beyond keyword matching
- ► Lightweight runs on consumer hardware
- ► Modular architecture for easy extension

Current Limitations:

- ► No personalization stateless by design
- ▶ Dataset scope 5.160 books vs. commercial scale
- ► Cold start problem for new books
- ► No feedback learning static recommendations

Future Research Directions:

- ► Hybrid approach: Combine content-based with collaborative filtering
- ▶ Better embeddings: Experiment with domain-specific models
- ► Privacy-preserving personalization: Local user preference learning
- ► Multi-modal features: Include cover images. genre embeddings

"Demonstrates that local-first ML - or using a more popular term - edge AI. can provide meaningful semantic recommendations without compromising user privacy or requiring cloud infrastructure."

8 / 14

Book Recommender using NLP

Critical Analysis & Future Directions

Future Research Directions

Privacy researcing by desire.

. Cold start problem for new books

- ► Lightweight runs on consumer hardware Better erebeddings: Expe domain-specific models
- Current Limitations: ► No personalization - stateless by design
 - a. Multi-modal features: Include cover impres
- b Dataset arose 5 MO books us, commercial arole

a. Muhid seconds: Combine content-based u

Conclusion



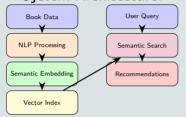
Research Question Answered:

How can a local ML model recommend books based on natural language descriptions?

Solution Implemented:

- ► Semantic embeddings with MiniLM transformers
- ► Vector similarity search using FAISS
- ► Zero-shot classification for categorization
- ► Privacy-first design fully local processing
- ► Key Contributions: Proof-of-concept that modern NLP enables practical, privacy-preserving recommendation systems

System Architecture:



Impact & Applications:

- Educational tool for privacy-aware ML
- ► Foundation for local-first recommendation systems
- Demonstrates transformer accessibility on consumer hardware

9 / 14

Book Recommender using NLP

—Conclusion

Research Question Answered:

- Semantic embeddings with MiniLM
 transformers
- ➤ Vector similarity search using FAISS ► Zero-shot classification for categorization
- ► Privacy-first design fully local processing ► Key Contributions: Proof-of-concept that



- ▶ Demonstrates transformer accessibility on consumer

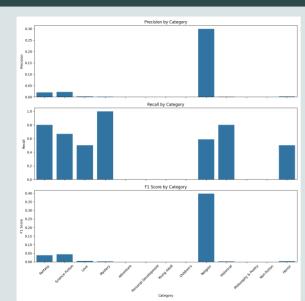
Thank you for your attention!

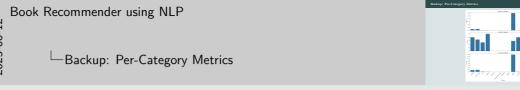


Questions?

Backup: Per-Category Metrics







Backup: Fallback Keywords

- c
 - 12
- Book Recommender using NLP

► Fallback und when sero-thet model coefficience was low ► Example Reywords: ○ Faterusy magic, witzerd, dragon ○ Science Fation, quoe, Al, dysteplia ○ Mystery: describe, class, edites

Backup: Fallback Keywords

- ► Fallback used when zero-shot model confidence was low
- ► Example keywords:
 - ▶ **Fantasy**: magic, wizard, dragon
 - Science Fiction: space, AI, dystopia
 - Love: romance, passion, relationship
 - Mystery: detective, clue, crime

Backup: Performance

- Book Recommender using NLP

- ▶ Embedding time per book: ≈ 2 ms (batch embedding)
- Query embedding: nr 50-200 ms ► FAISS search: < 10 ms
- ➤ UI render time: 1: 1-2 seconds (including image loading) ► All processing fully local on consumer-grade lastes

- - Backup: Performance

- ▶ Embedding time per book: \approx 2 ms (batch embedding)
- ▶ Query embedding: \approx 50-200 ms
- ► FAISS search: < 10 ms
- ▶ UI render time: \approx 1-2 seconds (including image loading)
- ► All processing fully local on consumer-grade laptop

Backup: Concrete Examples



14 / 14

Example Query 1:

"Books about artificial intelligence and ethics"

Top Results:

- ▶ "The Alignment Problem" Brian Christian
- ► "Life 3.0" Max Tegmark
- "Weapons of Math Destruction" Cathy O'Neil

Example Query 2:

"mystery novels with unreliable narrators"

Top Results:

- ► "Gone Girl" Gillian Flynn
- "The Girl on the Train" Paula Hawkins
- ▶ "In the Woods" Tana French

What This Demonstrates:

- ► Semantic understanding beyond keywords
- ► Abstract concept matching (ethics, unreliable narrators)
- ► Cross-genre discovery potential

Evaluation Challenges:

- ▶ No ground truth for "perfect" recommendations
- ► Subjective nature of book preferences
- Solution: Focus on semantic relevance rather. than prediction accuracy

Discussion Starters:

- ► How could one evaluate recommendation quality
- ▶ Books with sparse descriptions
- ► Could this approach work for other domains?

Book Recommender using NLP

Backup: Concrete Examples

Evamele Overv 1:

Ton Results:

► "Life 3.0" - Max Tegmark

* "Gone Gel" - Gillion Dunn The Gid on the Train" - Paula Muskins

■ "In the Woods" - Tana French

Evamele Overv 2:

What This Demonstrates "Rooks about artificial intelligence and

- . "The Alimoner Doblers" Brine Objects Evaluation Challenges: ► "Weapons of Math Destruction" - Cathy O'Nell
 - h No around teeth for "needers" promoundation Subjective nature of book preferences. - Solution: Enous on manuatic missource rather
 - Discussion Starters:

► How could one evaluate recommendation

- . Backs with some descriptions