#### Book Recommender using NLP

Carsten Lydekin

Zealand Business College

Oral Exam – AI and ML. 4th Semester

Book Recommender using NLP

Canaten Lydeklang
Zushand Bustone College
Oral Einem – AI and ML, 6th Scenester

## 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.

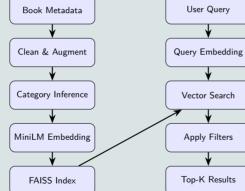
#### Architecture Overview



#### **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

► Data Pipeline: Clean & augment book metadata ➤ Category Inference: Zero-shot classification fallback rules ► Semantic Embedding: HIALDS sentence b Martor Search: \$4700 similarity marchine Key Innovation: Fully local semantic search

3 / 14

#### Dataset Exploration & Processing

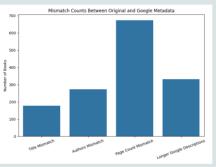


## **Data Quality Challenges:**

- ► 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: a. Started with 6 800 books from multiple sources

► Issues found: Missing author/category information
 Very short descriptions (< 9 words)
 Inconsistent categorization across sources. ► Final dataset: 5,160 high-quality books

► API perichment: OpenLibrary & Google Blocks

Data inconsistencies across sources require eveterratic cleaning and validation

#### Category Inference Strategy

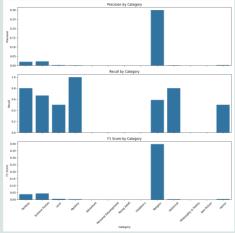


#### Two-Tier Classification Approach:

- ▶ 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
  - ▶ 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 8480-998

Confidence scoring for predictions . Extheric Dela-based insecond marchine

► Description length > 900 char

- ► Average coefidence > 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

Query: "space exploration novels" MiniLM Transformer Vector:  $\vec{v} \in \mathbb{R}^{384}$ FAISS L2 Distance Search Top-K Similar Books

End-to-end semantic search in < 200ms

Book Recommender using NLP

Semantic Embedding & Vector Search

Sentence Embedding with MiniLM: Model all-MailM-L6-92

► Input format:

"Trick ... Author ... Description

• Output: 200-dimensional vectors

► Advantage: Semantic similarity beyond keywords

Vector Search with FAISS:

Vector Search with FAISS:

> Index: 5,160 book on Needlings
> Search: L2 distance (exact search)
> Performance: < 10mm query time
> Local No. actumed descondencies

MiniLM Transformer

ds Vector: \$\vec{v} \in \mathbb{R}^{MB}\$

FAISS L2 Distance Search

Top-K Similar Books

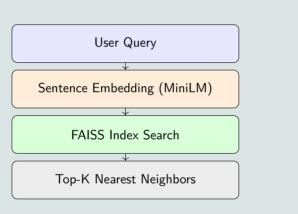
End-to-end semantic search in  $< 200 \mathrm{ms}$ 

### Vector Similarity Search with FAISS



- ► Used Facebook AI Similarity Search (FAISS) library
- ▶ Performs **L2 distance** search in embedding space
- ► Index built with:
  - 5160 book embeddings ( $\vec{v} \in \mathbb{R}^{384}$ )
- ► At runtime:
  - User query is embedded
  - ▶ Top-k nearest neighbors retrieved

▶ Results shown in UI ► Fully local, fast search



Book Recommender using NLP

└─Vector Similarity Search with FAISS

▶ Used Facebook All Similarity Search (EAISS) Union User Query Partners I 2 distance courts in embedding \_\_\_\_ Sentence Embedding (MiniLM) Index built with:
 5160 book embeddings (v ∈ R<sup>ths</sup>)
 Exact search (IndexFlatL2) FAISS Index Search User query is embedded
 Tap-is nearest neighbors retrieved
 Results shown in UI

- Bully local fort much

Top-K Nearest Neighbors

#### 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:

- ► Liebtweight runs on consumer hardwa domain-specific models
- ► Modular architecture for easy extension - Multi-modal featurer factors cover

#### ► No personalization - stateless by design

- ► Dataset scope 5.160 books vs. commercial
- . Cold start publics for one books ». No feedback leavains - static recommendations

irrages, genre embeddings

a Muhid sassanth: Combine content hase

#### Conclusion



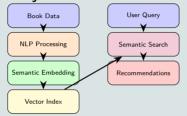
### **Research Question Answered:**

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

Book Recommender using NLP

—Conclusion

- Sensetic onheddings with MiniLM
   transformer ➤ Vector similarity search using FAISS
- ► Zero-shot classification for categorization
- ► Privacy-first design fully local processing



resident RLP establis practical,
privacy-preserving recommendation systems
consumer hardware



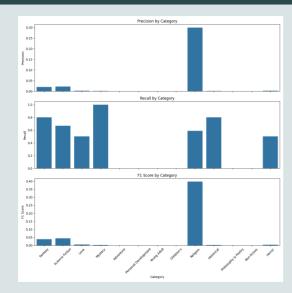
Questions?

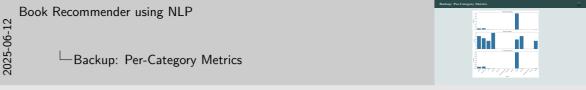
End of Presentation

Book Recommender using NLP

## Backup: Per-Category Metrics







## Backup: Fallback Keywords

- Book Recommender using NLP

► Fallback used when zero-shot model confidence was low Example knywords:
 Fantasy: magic, witand, dragon
 Science Fiction: space, Al, dystopia
 Low: normano, pastion, estationship
 Mystery: distriction, close, clinic

- ► 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: Fallback Keywords

## Backup: Performance

- c
- Book Recommender using NLP
  - NLP

- ➤ Embedding time per book: 10 2 res (batch embedding)
  ➤ Query embedding: 10 50-200 res
- FAISS search: < 30 ms
  Ull render time: m 1-2 seconds (including image leading)
  All processing fully local on consumer-grade laptop

Backup: Performance

- ► Embedding time per book: ≈ 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



## **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 "Books about artificial intelligence and

Backup: Concrete Examples

### What This Demonstrates

Top Results: ▶ "The Alignment Problem" - Brian Christia ► "Life 3.0" - Max Teemark

Top Results:

a "Good Girl" - Gillion Diese

Evaluation Challenges: . "Winness of Math Destruction"

Discussion Starters: ► How could one evaluate recommendat \* "The Girl on the Train" - Paula Hawkins

► Books with sparse descriptions Could this approach work for other