Emoji Book Recommender

Shivani Ramesh, Isabelle Reilly, Lillian Ehrhart

Agenda

- Introduction
- Use Case
- NLP Task: Information Retrieval
- Emoji data preparation & keyword annotation
- Preparing the dataset of books
- Walkthrough of the pipeline
- Model Evaluation
- Demo
- Conclusion and Limitations
- Future Directions

Introduction

- Problem definition: How can we create a unique, interesting way to recommend books to the user?

- Details considered:
  - How can we make the program accessible and easy to use?
  - What should we consider when making recommendations?
    - Genre? → broader topic words
  - What are potential limitations of the approach?
    - Relying on book descriptions
  - How might we evaluate such a model?
    - Subjectivity of relevance scoring

Use Case: Why this project?

- In some areas, a user might not have fully articulated keywords in mind for a search
- One case: looking for something new to read!
    - Users might not know exactly what topic they're looking for
    - They might not know exactly what genre or age range they want
    - They may not have a specific title or author in mind
    - ...But they tend to have at least light preferences in some of these areas

- Our idea is to experiment with a fun, engaging method of input in a context where precision may not be the goal for the user

NLP Task: Information Retrieval

- Open-minded use case
- Distinct from in-class discussion of searching for a restaurant

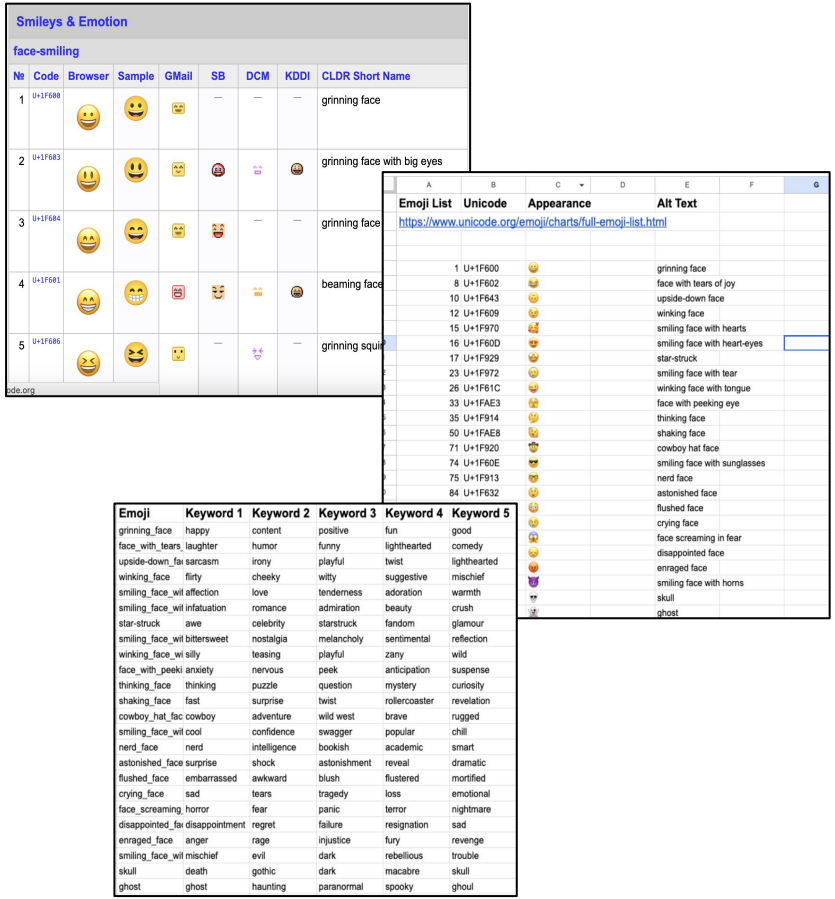
Other differences:

- User does not input an actual keyword — more interpretation on the part of the program
- Weighting and doubling emoji input

Datasets and Annotation

Emoji Dataset and Annotation

- Started with Unicode emoji list
- Pared down and added to spreadsheet, eliminated emoji that were:
  - Redundant (multiple smileys)
  - Not widely applicable in context (horoscope symbols)
- Added five keywords to describe each emoji
  - Started with list of Chat GPT-generated keywords, then edited and replaced with our own tailored keywords
- Finally, formatted into TSV file—ready to work with!



Smileys & Emotion

face-smiling

№	Code	Browser	Sample	GMail	SB	DCM	KDDI	CLDR Short Name
1	U+1F600				—	—	—	grinning face
2	U+1F603							grinning face with big eyes
3	U+1F604					—	—	grinning face
4	U+1F601							beaming face
5	U+1F606				—		—	grinning squi

Emoji List	Unicode	Appearance	Alt Text
https://www.unicode.org/emoji/charts/full-emoji-list.html			
1	U+1F600		grinning face
8	U+1F602		face with tears of joy
10	U+1F643		upside-down face
12	U+1F609		winking face
15	U+1F970		smiling face with hearts
16	U+1F60D		smiling face with heart-eyes
17	U+1F929		star-struck
23	U+1F972		smiling face with tear
26	U+1F61C		winking face with tongue
33	U+1FAE3		face with peeking eye
35	U+1F914		thinking face
50	U+1FAE8		shaking face
71	U+1F920		cowboy hat face
74	U+1F60E		smiling face with sunglasses
75	U+1F913		nerd face
84	U+1F632		astonished face
			flushed face
			crying face
			face screaming in fear
			disappointed face
			enraged face
			smiling face with horns
			skull
			ghost

Emoji	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5
grinning_face	happy	content	positive	fun	good
face_with_tears_	laughter	humor	funny	lighthearted	comedy
upside-down_fa	sarcasm	irony	playful	twist	lighthearted
winking_face	flirty	cheeky	witty	suggestive	mischief
smiling_face_wi	affection	love	tenderness	adoration	warmth
smiling_face_wi	infatuation	romance	admiration	beauty	crush
star-struck	awe	celebrity	starstruck	fandom	glamour
smiling_face_wi	bittersweet	nostalgia	melancholy	sentimental	reflection
winking_face_wi	silly	teasing	playful	zany	wild
face_with_peeki	anxiety	nervous	peek	anticipation	suspense
thinking_face	thinking	puzzle	question	mystery	curiosity
shaking_face	fast	surprise	twist	rollercoaster	revelation
cowboy_hat_fac	cowboy	adventure	wild west	brave	rugged
smiling_face_wi	cool	confidence	swagger	popular	chill
nerd_face	nerd	intelligence	bookish	academic	smart
astonished_face	surprise	shock	astonishment	reveal	dramatic
flushed_face	embarrassed	awkward	blush	flustered	mortified
crying_face	sad	tears	tragedy	loss	emotional
face_screaming	horror	fear	panic	terror	nightmare
disappointed_fa	disappointment	regret	failure	resignation	sad
enraged_face	anger	rage	injustice	fury	revenge
smiling_face_wi	mischief	evil	dark	rebellious	trouble
skull	death	gothic	dark	macabre	skull
ghost	ghost	haunting	paranormal	spooky	ghoul

Emoji Keywords

Short-text | First Keyword | Other Keywords

student	student	learning	school	education	study
teacher	teacher	education	school	instruction	classroom
judge	judge	law	justice	court	authority
farmer	farmer	agriculture	earth	gardening	nature

Short-text label: fed into back-end with Python emoji library, stored in keyword matrix

First keyword: parallel to the short-text, the primary keyword and/or name of the emoji

(after initial testing, we decided to weight this more heavily to get more topical books)

Other keywords: related topics, settings, and descriptive words

Where did we get the book information? Trial and error

Try 1:

- Google Books
- Open Library
- Internet Archive

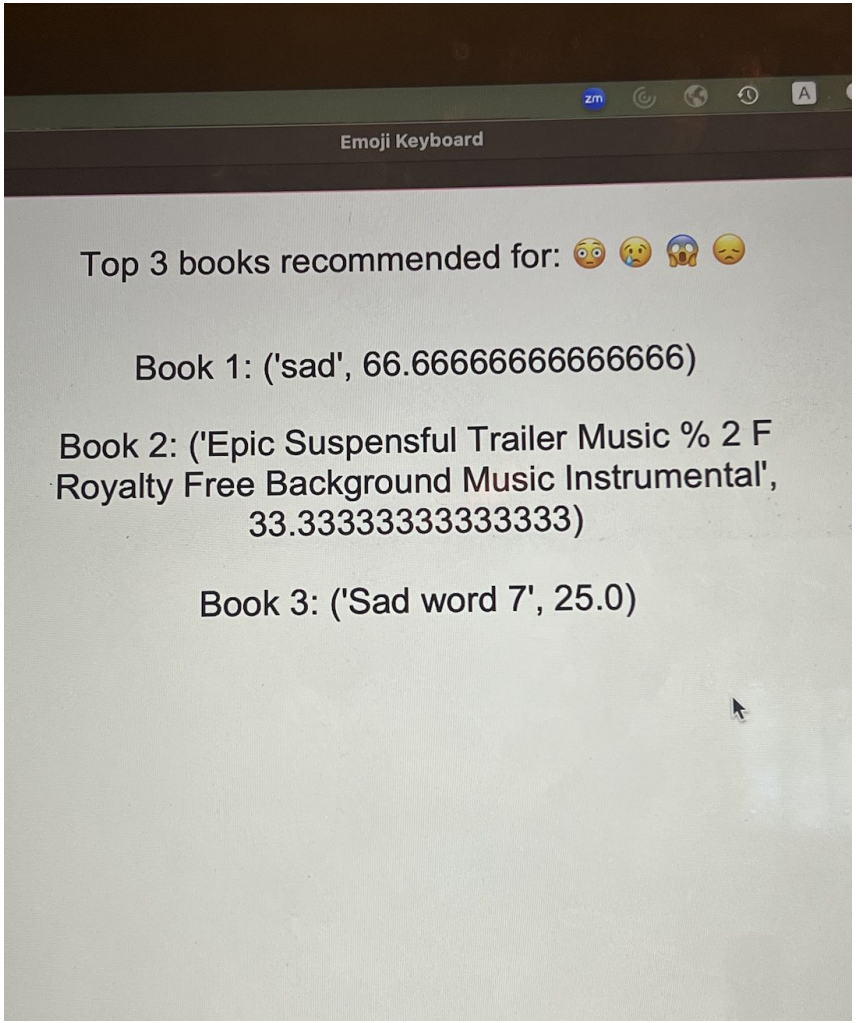
Queried using keywords derived from emoji inputs → 43,000 books

Stored data includes:

- Title, authors, publisher, description, etc.

Issues:

- Internet Archive included different media types - not just books
- Open Library was missing book descriptions


Emoji Keyboard
Top 3 books recommended for: 😳 😢 😱 😞
Book 1: ('sad', 66.66666666666666)
Book 2: ('Epic Suspensful Trailer Music % 2 F Royalty Free Background Music Instrumental', 33.33333333333333)
Book 3: ('Sad word 7', 25.0)

Where did we get the book information? Trial and error

😀

Try 2: Kaggle!

- Kaggle data scraped from wonderbk.com (an online book store) which contains details of 100,000+ books

Stored data includes all the same information:

- Title, authors, publisher, description, etc.

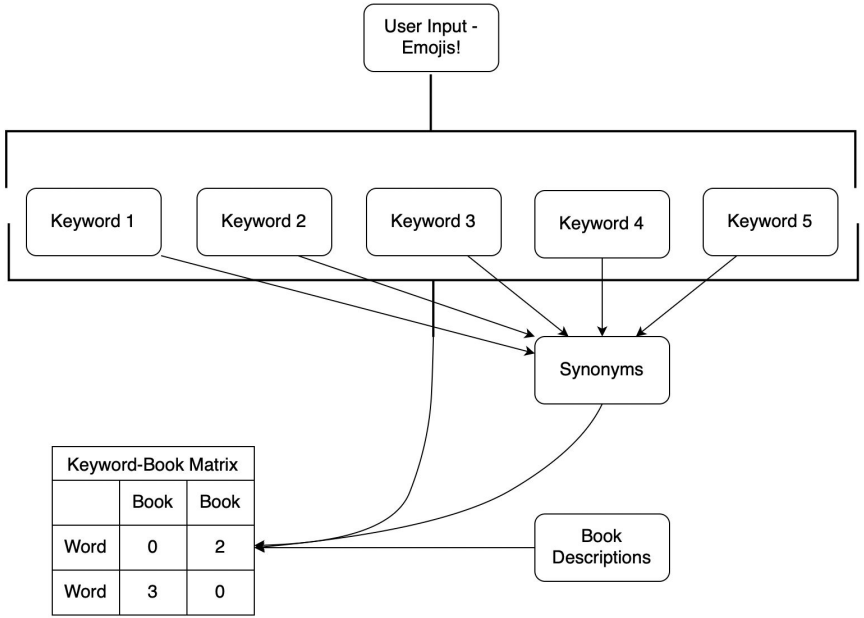
Filtered out all books which didn't have a description field

Easy to parse and extract to make term-document matrix

Book Dataset: Incorporating into the Pipeline

1. Book dataset file → book descriptions

2. Into the pipeline! → Use book descriptions to count keywords and create a term-document matrix...



Keyword-Book Matrix		
	Book	Book
Word	0	2
Word	3	0

Our Pipeline

- Building the keyword-book matrix using Kaggle data
- User Interface
- Query Processing

Building the Matrix

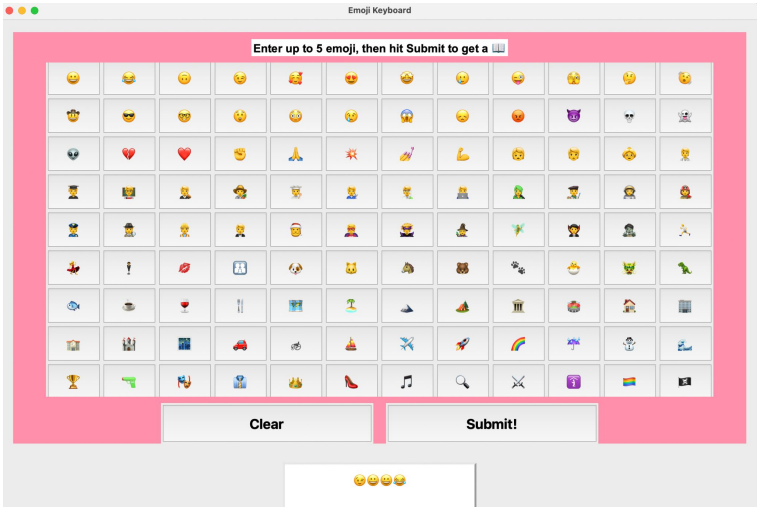
Book Title and Author

Keyword	Animal Farm	Twilight	War & Peace	Dracula
barnyard	2	0	0	0
vampire	0	5	0	2
Russia	1	0	2	1

- Large dataset led to complications
- Term-document matrix stores books per keyword (including synonyms)
  - Synonym support from NLTK WordNet
  - Normalized to adjust for description length (with raw counts, the search privileged books with longer descriptions)
  - Filter out books w/ empty descriptions

User Interface & Main Loop

tkinter GUI:



- As simple as possible: clear instructions and visual design
- Interactive keyboard with display (Python emoji library)
- Simple buttons: option to reset input and submit
- No need to interact with terminal to operate
    - Once user submits their emoji selection, keyboard freezes and UI changes to a results page

Processing Between User Query and Matrix

- Emoji short-text associated to keywords
- Search through matrix file with user query
  - Matrix file is generated separately beforehand to cut down runtime
- Weight keywords
  - Book score = (count_book) *(count_query)
  - Prioritizes keywords with high frequency in both book and query
- Return sorted list of book scores

Evaluation & Conclusions

Model Evaluation 😭

- Not always a single “correct” answer → hard to define accuracy
- Subjectivity: Relevance and accuracy depends on user interpretation
- Lack of labeled data for relevance → ideally we would get human annotators

Software Demo!

Limitations

- Difficulty in evaluating the model
  - True success is somewhat subjective (i.e. did they like the book?)
- Relying on descriptions of books that we didn't write ourselves → sometimes descriptions are not robust and/or topical
  - Need databases w/ descriptions, which may exclude certain kinds of books
- User can upload their own book database, but it's not completely "plug and play"

Future Directions

Generalizable to other media

- Movies? Shows? Songs? Poems? Pieces of art?

Filters for stronger user preferences

- Fiction vs. Nonfiction
- Length
- Excluding certain content

More accessibility features for the UI

- Used descriptive labels and buttons, but would ideally add more screen readers support
- Toggles for high-contrast emoji display
- Toggles to adjust the size of each emoji key for easier reading

Quality testing on a variety of devices

Bibliography

Dataset:
https://www.kaggle.com/datasets/elvinrustam/books-dataset?select=BooksDatasetClean.csv

Recent Research:
https://www.nature.com/articles/s41598-025-92286-0

- Presents a novel sentiment analysis approach that effectively incorporates emojis and emoticons in processing US airline tweets, using machine learning classifiers and BERT, achieving 92% accuracy—9% higher than state-of-the-art models.
https://www.sciencedirect.com/science/article/pii/S0957417424010194
- Presents a Systematic Literature Review (SLR) of deep learning-based text summarization techniques (both extractive and abstractive) from 2014 to 2023, analyzing 73 studies to explore models, input representations, training strategies, evaluation metrics, datasets, and challenges in the field.
https://www.researchgate.net/publication/334765586_Automated_Genre_Classification_of_Books_Using_Machine_Learning_and_Natural_Language_Processing
- Proposes a machine learning-based genre classification method for books by transforming text into a feature matrix, reducing dimensionality using WordNet and Principal Component Analysis (PCA), and applying an AdaBoost classifier, achieving 92.88% accuracy by incorporating unlabeled data.

Thank you!