Movie Finder

Erdos Data Science Bootcamp Project

Team Supermassive Blackhole

Anna Brosowsky*, Sayantan Khan*,
Nancy Wang*, Ethan Zell*, Yili Zhang*

*University of Michigan

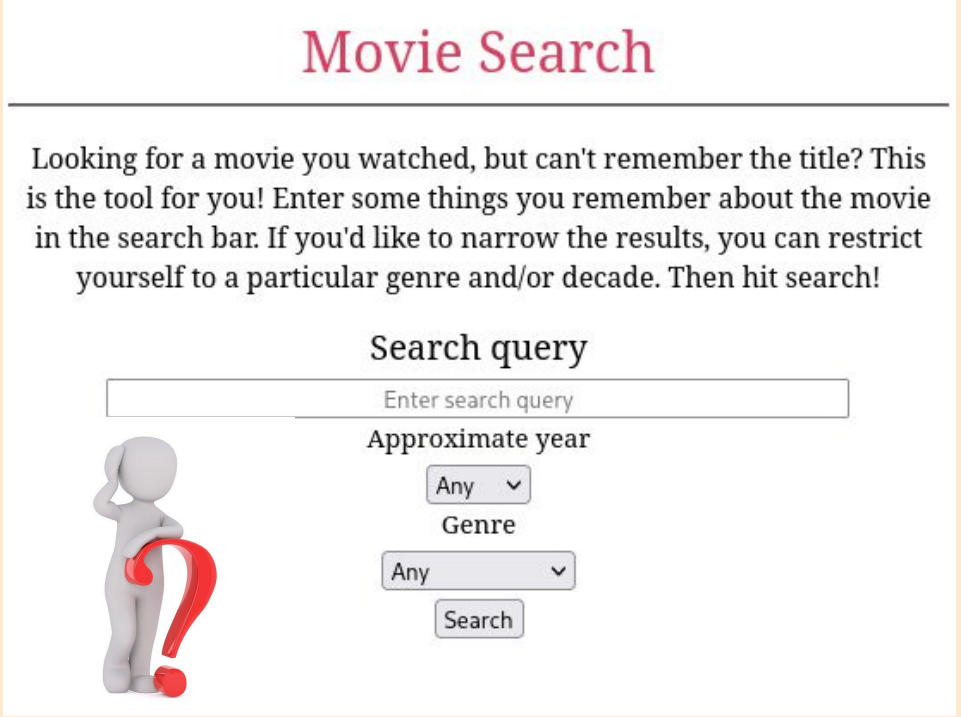
Overview

Goal: build a movie search app which will give the title of the movie using what the user remembers about the plot.

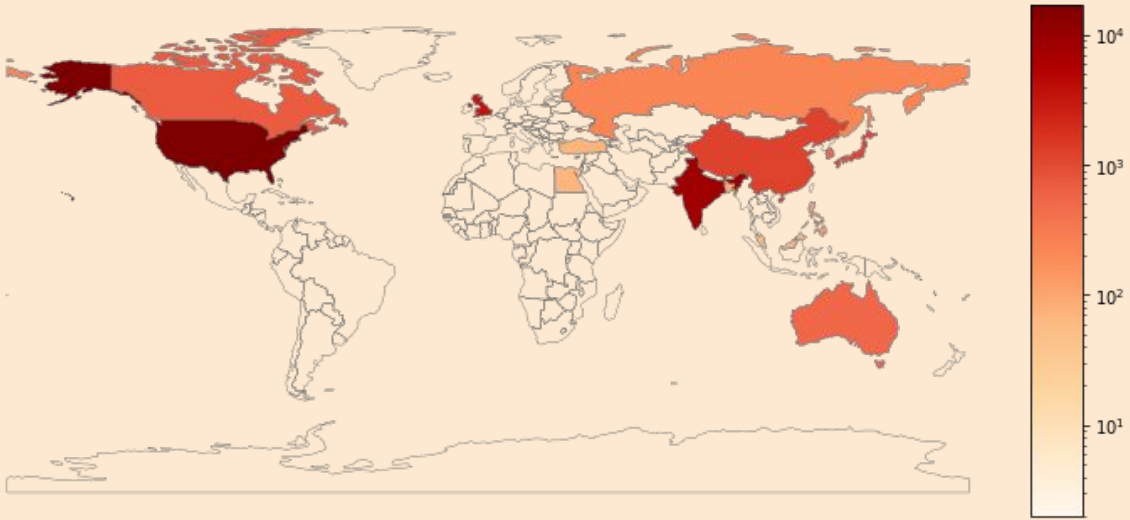
Data: we combined three existing data sets (Kaggle “Wikipedia Movie Plots” [1], Kaggle “The Movies Dataset” [2], and “CMU Movie Summary Corpus” [3]).

For testing, we also scraped the IMDB website to collect user-contributed plot summaries.

Spoiler:



Movie Production Location

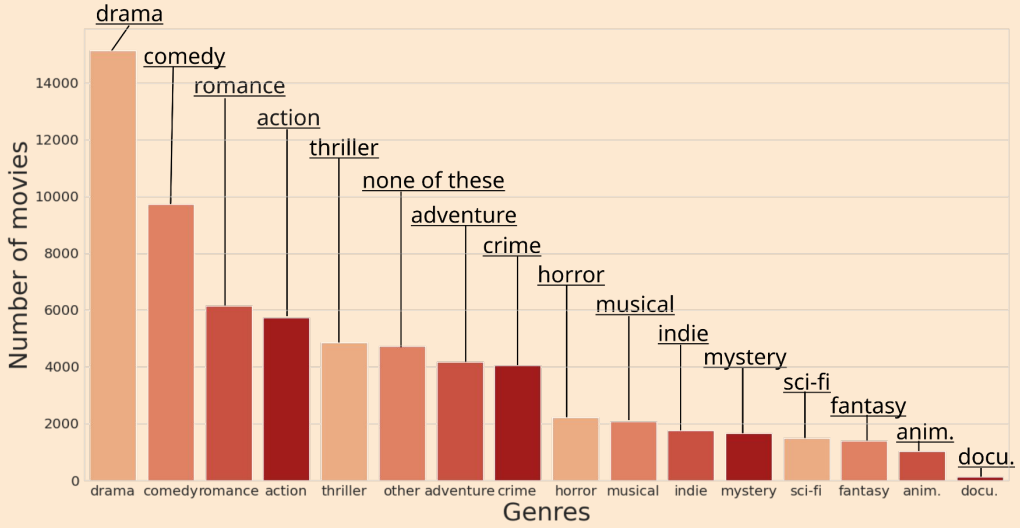


The modified data set contains more than 30000 movie records. These movies are released between 1901 and 2017.

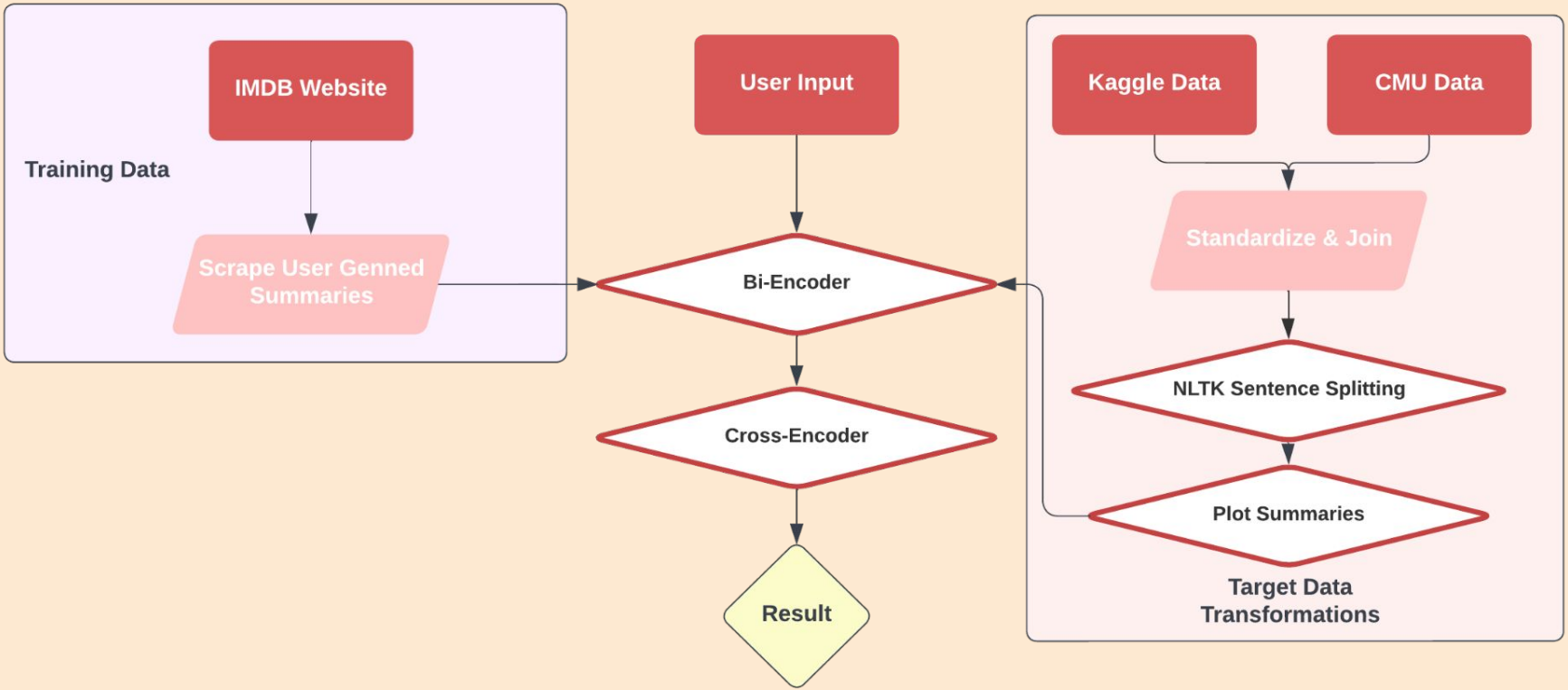
In the modified data set:

- Movies made in **15 countries**; 50% of the movies are American movies.
- **47%** of the movies are released **after 1990**.
- **Drama** is the most popular genre.

Year & Genre Data



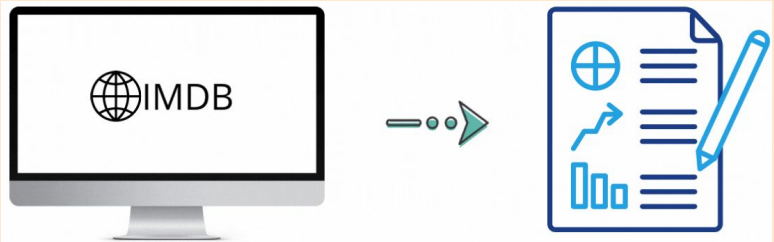
NLP Pipeline Overview



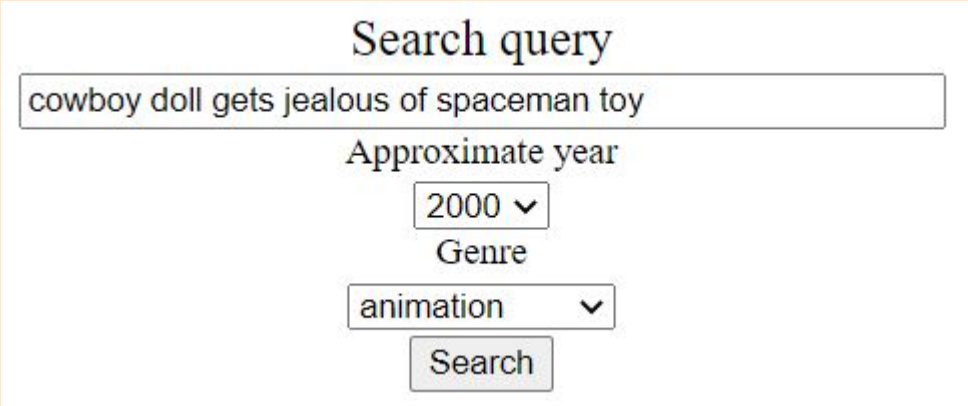
Intake Pipeline

1. Took advantage of IMDB's url structure that is organized around IMDB ID, a unique movie identifier.

2. **Web scraped** 2 user written summaries (when available) from IMDB site.



User input consists of a vague movie description, an approximate year, and genre.



Model Building

1. Break up Wiki plot summaries into chunks of ≤ 256 tokens using **NLTK's sentence tokenizer[6]**.

2. Semantically embed the plot summaries into a high dimensional vector space using a **BERT-based transformer [4]**.

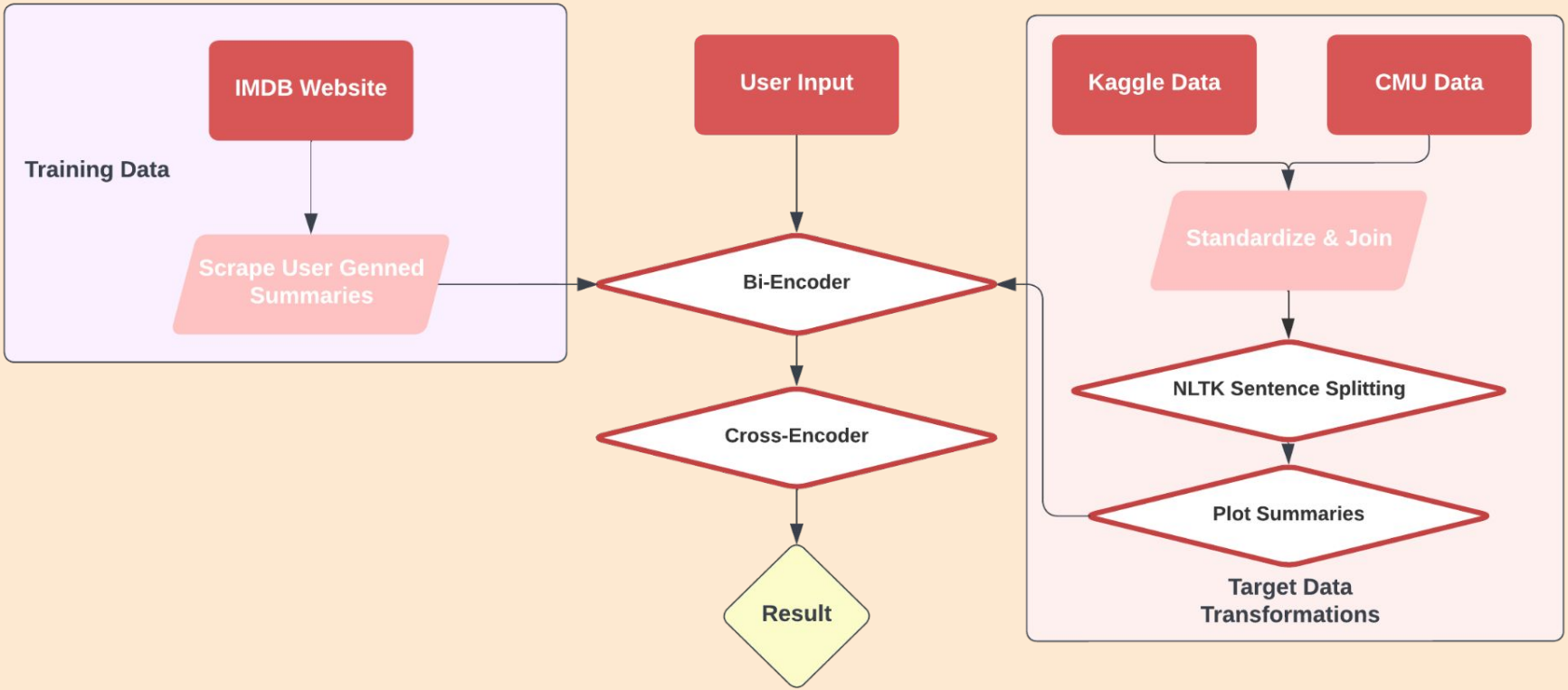
Classification Pipeline

1. **Semantically embed** user query into same high dimensional vector space.

2. Find the 100 closest plot summaries (using **cosine-similarity**) to the user query.

3. Compute similarity scores between query and each of the top 100 plot summaries using a **MiniLM-based transformer**, and return the top 10 results.

NLP Pipeline Overview



Results

Link to the web app:
http://jupyter.sayantankhan.io/search

Accuracy:

- **Cosine similarity** on the embedded gives 75% accuracy on the IMDB query dataset.
- Reranking using the **cross encoder** boosts the accuracy to 84%.

Query from example: "*cowboy doll gets jealous of spaceman toy*", Approximate year: 2000, Genre: Animation.

Future Directions

- Incorporate **filtering** using cast/director info.
- Use genres in a **weighted** way instead of as a filter.
- Let users **continually train** the model (a “this is my movie!” button).
- Find **more** representative test **data** to increase the diversity of movies in database.
- Incorporate **other models** that use significantly different techniques & average scores.

References

[1]: Kaggle “Wikipedia MoviePlots” https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots

[2]: Kaggle “The Movies Dataset” https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

[3]: “CMU Movie Summary Corpus” http://www.cs.cmu.edu/~ark/personas/

[4]: *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*, Nils Reimers, Iryna Gurevych

[5]: *Scalable Zero-shot Entity Linking with Dense Entity Retrieval*, Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, Luke Zettlemoyer

[6] NLTK Package https://www.nltk.org/

Thank you!