INNOMATICS
RESEARCH LABS

Project On:

Enhancing Search Engine Relevance for Video Subtitles

Team: Madhu Priya and Thanvi Reddy

Objective

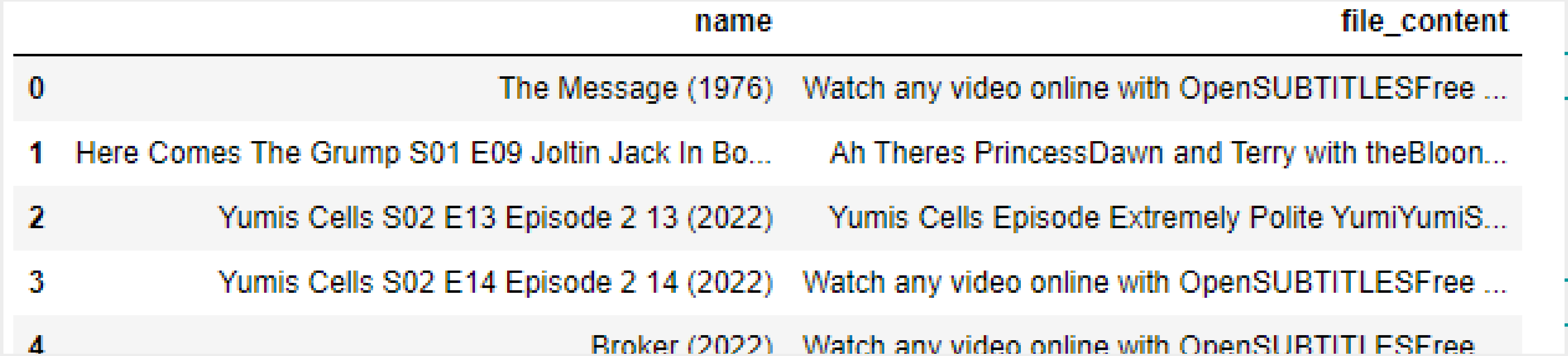
In today's fast-paced digital landscape, finding relevant content within videos, especially for subtitle users, can be challenging. Our solution aims to change that.
We've developed a cutting-edge search algorithm specifically tailored for video subtitles, leveraging state-of-the-art natural language processing and machine learning techniques. By understanding the context and meaning behind user queries, we're enhancing the search experience like never before.
In this presentation, we'll outline our methodology, from data ingestion to document retrieval, highlighting the advantages of semantic search over traditional keyword-based approaches. Ultimately, the goal is to improve search accuracy and relevance for video subtitles using natural language processing and machine learning methods.

Step 1: Connecting to SQLite

1. **Retrieve Table Information with SQLite**: Used **sqlite3** to connect to the SQLite database and execute PRAGMA to fetch column details from the 'zipfiles' table.
2. **Read Data into DataFrame**: Loaded data from the 'zipfiles' table into a pandas DataFrame (**df**) using **pd.read_sql_query()**.
3. **Extract Subtitle Content from ZIP Files**: Defined a function (**decode_method**) to extract subtitle content from binary ZIP data, utilizing **zipfile** module for decompression and decoding.
4. **Process Binary Data**: Accessed binary data from a specific DataFrame row and passed it to **decode_method** to extract subtitle content.
5. **Display Subtitle Content**: Printed the extracted subtitle content (assumed Latin-1 encoding) to verify the extraction process.

Step 2: Data Cleaning

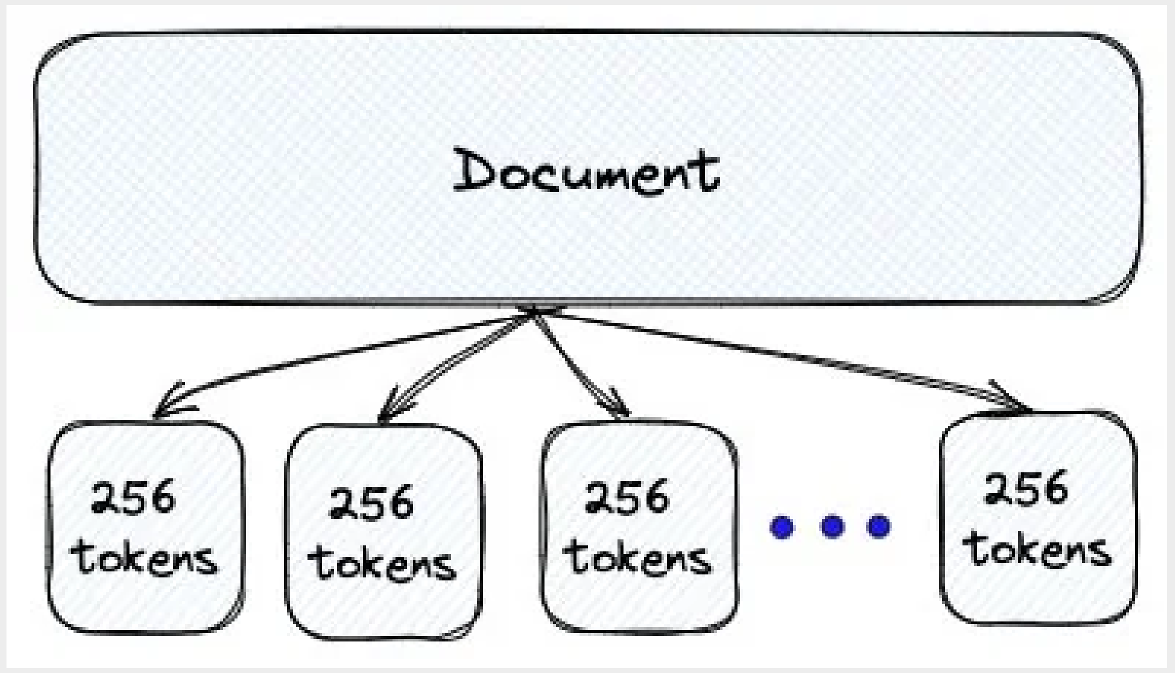
1. **Timestamp and Special Character Removal:** Eliminated timestamps, \r, \n tags, and numerical values embedded within the video subtitle content.
2. **Column Selection:** Retained only the 'title' or movie name column and the 'file_content' column containing the cleaned video subtitles.
3. **Proper Movie Name Formatting:** Ensured that movie names were cleaned and formatted appropriately for consistency and readability.



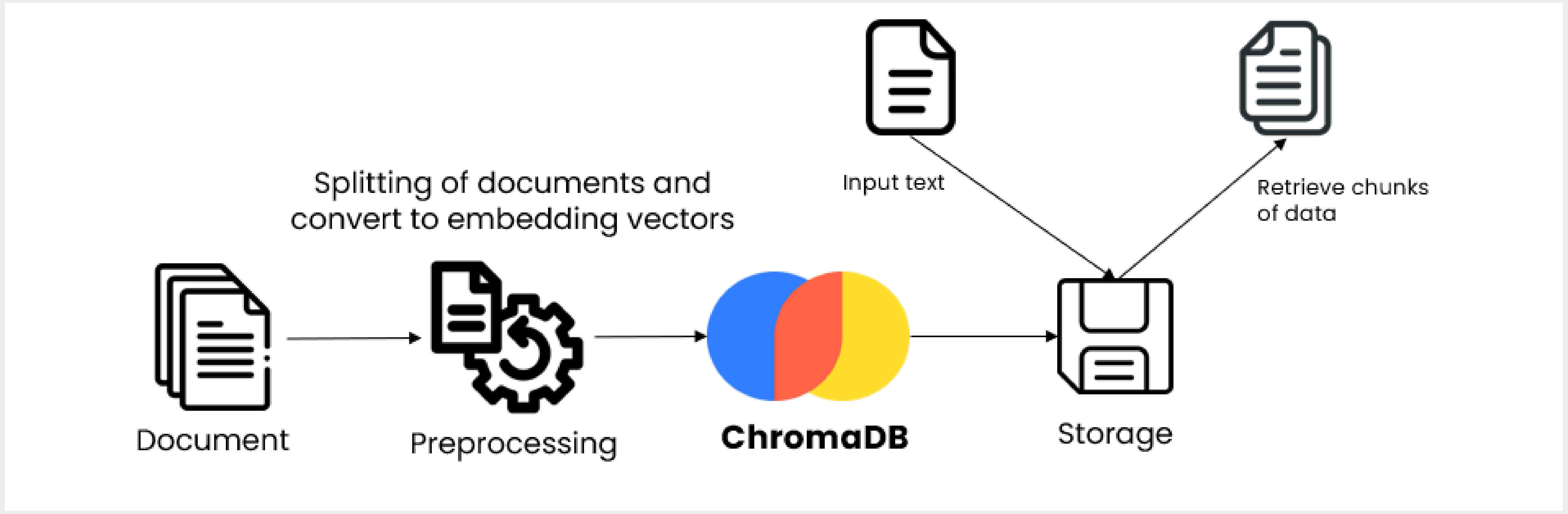
	name	file_content
0	The Message (1976)	Watch any video online with OpenSUBTITLESFree ...
1	Here Comes The Grump S01 E09 Joltin Jack In Bo...	Ah Theres PrincessDawn and Terry with theBloon...
2	Yumis Cells S02 E13 Episode 2 13 (2022)	Yumis Cells Episode Extremely Polite YumiYumiS...
3	Yumis Cells S02 E14 Episode 2 14 (2022)	Watch any video online with OpenSUBTITLESFree ...
4	Broker (2022)	Watch any video online with OpenSUBTITLESFree

Step 3: Chunking

- Chunking is breaking a big piece of data into smaller parts for easier handling. We use it to save memory, speed up processing, and work with large datasets more efficiently. It's like dividing a big task into smaller, more manageable pieces.
- To convert each chunk into tokens, we used a technique called tokenization. Tokenization breaks a text into individual tokens.
- We loaded the 'all-mpnet-base-v2' SentenceTransformers model. You can use any other BERT-based model depending on your requirements.
- We iterate over each chunk, encode it using the BERT model, and store the resulting embeddings in the bert_embeddings list.



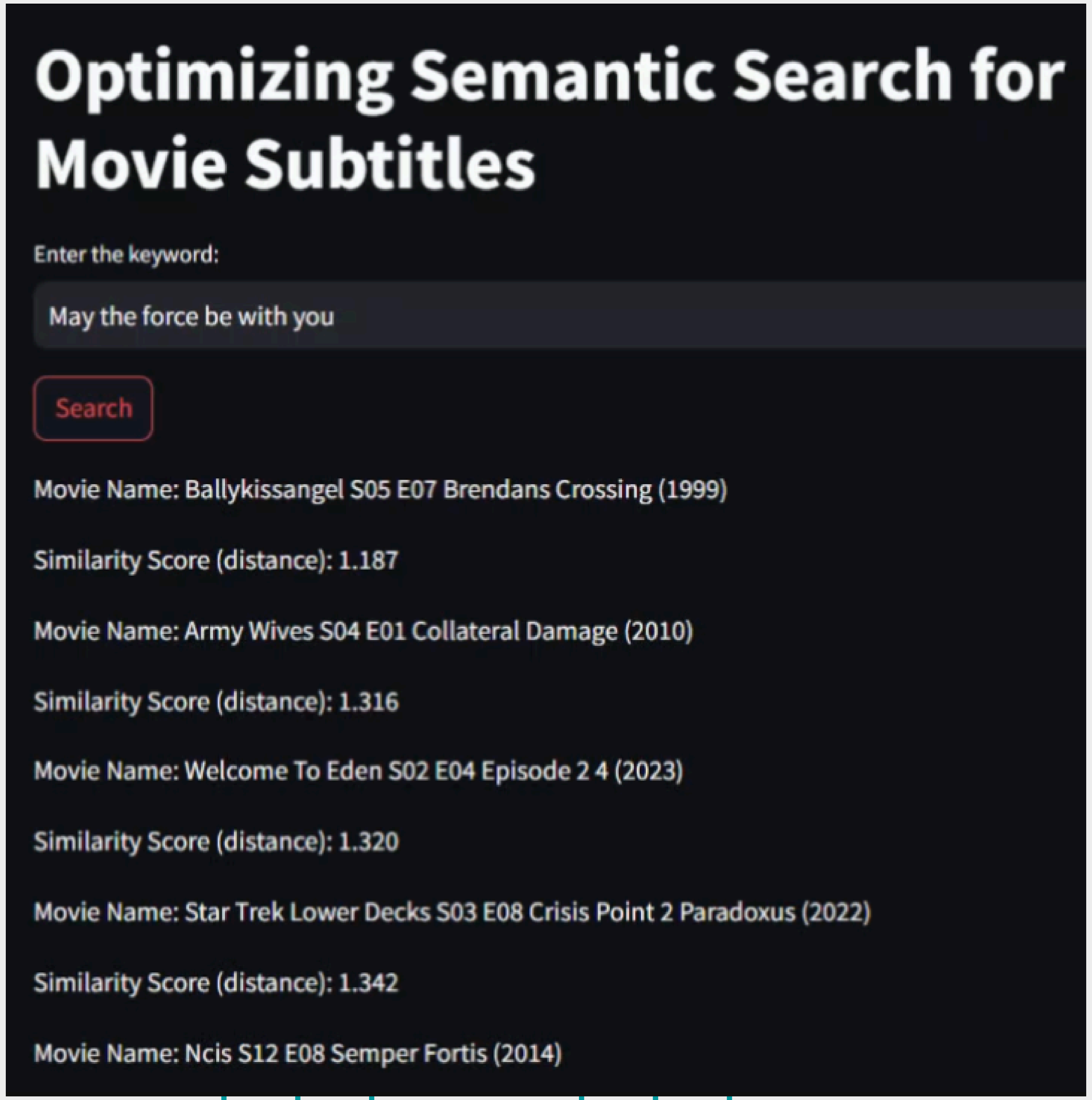
Step 4: ChromaDB & Embeddings setup



- VectorDB (ChromaDB) database is created with PersistentClient for storage and retrieval of embeddings.
- An embedding function is defined using the SentenceTransformer model.
- Collection "eng_subtitles_collection" stores documents, embeddings, and metadata.
- The search function retrieves the top 10 similar documents based on cosine similarity.

Step 5: Search functionality implementation

- Developed a user-friendly application using Streamlit.
- User inputs a movie query into the interface.
- Query is encoded using the same model as the dataset.
- Implemented cosine similarity for efficient distance calculation. Cosine similarity finds the closest embeddings to the query.
- Displays corresponding movie names based on the top 10 closest embeddings to the user.


Optimizing Semantic Search for Movie Subtitles

Enter the keyword:

May the force be with you

Movie Name: Ballykissangel S05 E07 Brendans Crossing (1999)

Similarity Score (distance): 1.187

Movie Name: Army Wives S04 E01 Collateral Damage (2010)

Similarity Score (distance): 1.316

Movie Name: Welcome To Eden S02 E04 Episode 2 4 (2023)

Similarity Score (distance): 1.320

Movie Name: Star Trek Lower Decks S03 E08 Crisis Point 2 Paradoxus (2022)

Similarity Score (distance): 1.342

Movie Name: Ncis S12 E08 Semper Fortis (2014)