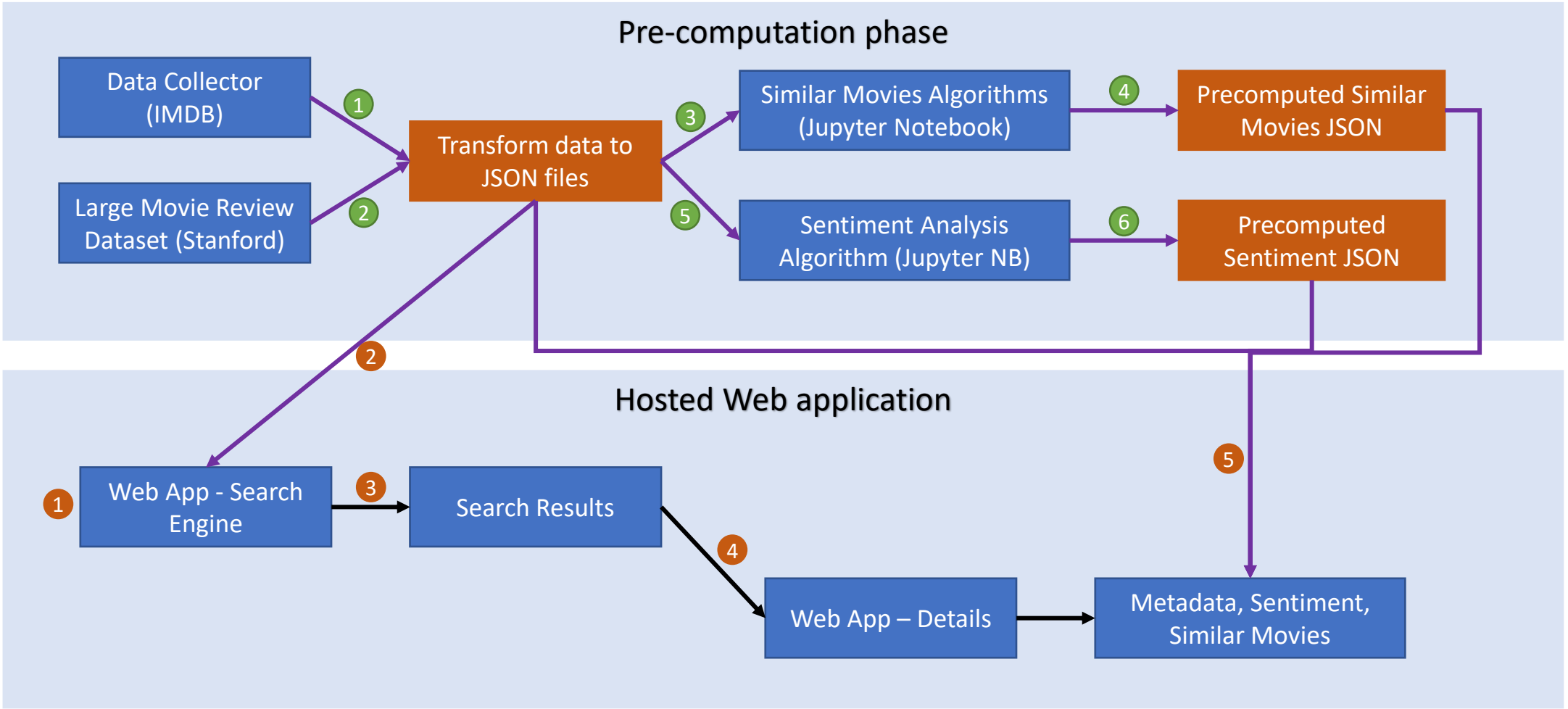
Intelligent Movie Search: Goals

- Team Members
  - Danh Nguyen NetID: danhn2@illinois.edu
  - Justin Quach NetID: jmquach2@illinois.edu
  - Tik On (Johnson) Lui NetID: tlui2@illinois.edu
  - Vamshidhar Kommineni NetID: komminen@illinois.edu
- Goals
  - Learn about and use the algorithms and techniques from CS 410 in a simple real-world application
  - Build an application that can be extended easily by other students in future semesters
  - Find some additional areas of learning both in information retrieval and general CS

Intelligent Movie Search: Approach

- Choose a rich data set: Stanford Movie Reviews Large Dataset + IMDB Metadata
- Design the use cases, data needed, etc.
  - Search Engine (Names and Metadata)
  - Review Sentiment
  - Similar Movies
- Data Collection and transformation: Fetch, schematize and store for later steps
- Experiment with the algorithms/methods, evaluation and produce the data/modules for the application:
  - Choice of three rankers for search engine (BM25, Jelinek-Mercer, Dirichlet Prior)
  - Pre-computed sentiment analysis (Vader)
  - Choice of methods for similar movies (BERT, Word2Vec, TF-IDF)
- Build a simple web application as a front-end for the Information retrieval techniques
  - Stretch: Host web application for easy use/grading/experimentation – Hosted on Azure and pythonanywhere

Intelligent Movie Search: Architecture

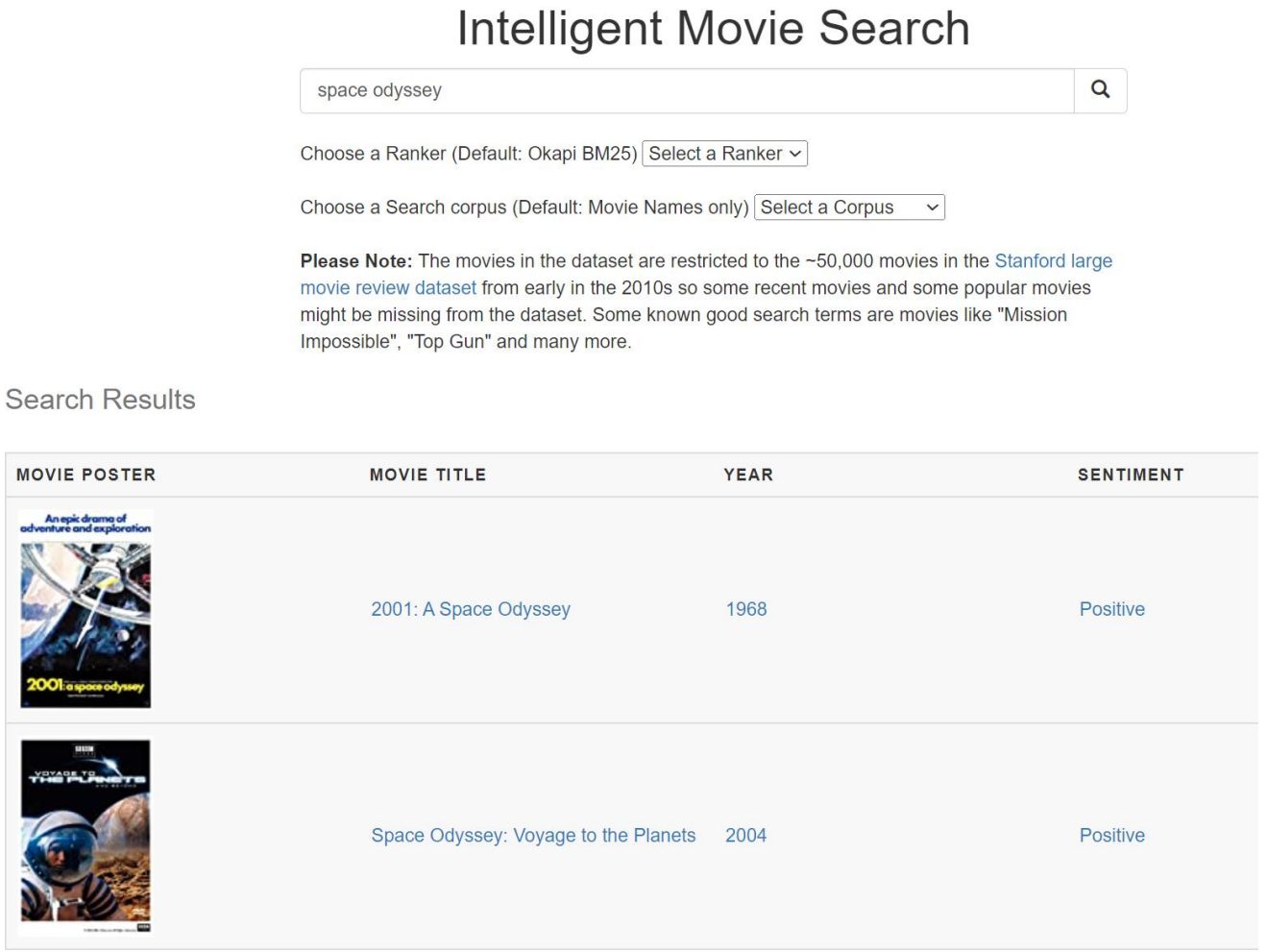


Intelligent Movie Search: Installation

- **Installation for usage**: None. Visit one of these two websites:
  - http://cs410djtv.azurewebsites.net
  - http://komminen.pythonanywhere.com
  - **Note:** Both websites are identical. First few uses/clicks after a while can be slow because of container reloads, warming up data, etc. on the hosting platform.
- **Installation for extension** (not needed for peer grading): Fully documented in the report
  - Git clone https://github.com/komminen/CS410_DJTV.git
  - Setup anaconda, python3.5, pip install --r requirements.txt
  - Run with “python main.py” and open up the browser to localhost

Intelligent Movie Search: Demo/Testcases

- Search Engine **test case**: Default ranker, default corpus (movie names)
  - Try a movie like “Space Odyssey” in the search box
- Search Engine **test case**: Choose a ranker or search corpus from the drop down and re-run the search
  - E.g. Try “Meryl Streep” with default corpus and there are no results.
  - With “All movie metadata” for corpus, you see movies that she acted in
- **Note:** The movie list matches the ~50K movies in the Stanford movie dataset so it isn’t exhaustive



Intelligent Movie Search: Demo/Testcases

- **Test case** for metadata, sentiment, similar movies
- Click on the first result from the search for “space odyssey” to see details about the movie http://cs410djtv.azurewebsites.net/detail/0062622/
- **Metadata:** Metadata fetched from IMDB is shown including a link to IMDB to learn more about the movie
- **Sentiment:** There were 30 reviews in the movie review research dataset, with 27 classified as positive by the Vader algorithm
- **Similar Movies:** Scroll down to see similar movies all with a space theme using BERT. You can change to Word2Vec or TF-IDF and update to see completely different sets of similar movies

Movie Details

ATTRIBUTE	VALUE
Movie Poster	
Name (Click name for link to IMDB)	2001: A Space Odyssey
Year Released	1968
Length (minutes)	149
Rating	8.3
Review Sentiment	**Positive** 30 Reviews Analyzed sentiment: 27 Positive | 1 Negative | 2 Neutral
Genre(s)	Adventure, Sci-Fi,
Summary	After uncovering a mysterious artifact buried beneath the Lunar surface, a spacecraft is sent to Jupiter to find its origins - a spacecraft manned by two men and the supercomputer H.A.L. 9000.