

## Files in the Dataset

### 1. yelp\_academic\_dataset\_business.json

- **1.2M+ businesses**
- **Key fields:**
  - business\_id: Unique ID
  - name: Business name
  - address, city, state, postal\_code, latitude, longitude
  - stars: Average star rating (float)
  - review\_count: Number of reviews
  - is\_open: Open status (1 = open)
  - categories: Comma-separated categories (e.g., "Japanese, Sushi Bars, Restaurants")
  - attributes: Nested dict with metadata (e.g., "WiFi", "Parking", "Good for kids")

#### Use in RL:

- To enrich item context (e.g., categories, popularity)
  - To support action selection features
- 

### 2. yelp\_academic\_dataset\_review.json

- **8.6M+ reviews**
- **Key fields:**
  - review\_id
  - user\_id, business\_id: Relational links
  - stars: Actual rating (1–5 stars)
  - date: Timestamp of the review
  - text: Freeform review text (not always used in structured models)
  - useful, funny, cool: Count of feedback from other users

#### Use in RL:

- Forms the core of (state, action, reward) triplets

- The rating becomes reward
  - Enables sorting by time to form sequences
- 

### 3. yelp\_academic\_dataset\_user.json

- **2.1M+ users**
- **Key fields:**
  - user\_id, name
  - review\_count
  - yelping\_since: Registration date
  - friends: Comma-separated user\_ids
  - useful, funny, cool: Aggregate votes
  - fans, elite: Popularity/social influence indicators
  - average\_stars: Avg. rating given

#### Use in RL:

- Used to enrich the **state** with user profile (average\_stars, review\_count, friend count)
  - Can personalize action-value function
- 

### 4. yelp\_academic\_dataset\_checkin.json

- **Check-in records per business**
- **Key fields:**
  - business\_id
  - date: Comma-separated timestamps (e.g., "2018-01-01 09:00:00, 2018-01-02 18:00:00")

#### Use in RL:

- Extract time-based popularity (morning / afternoon / evening visits)
  - Serves as context feature for action (business popularity signal)
- 

### 5. yelp\_academic\_dataset\_tip.json

- **Short tips (mini reviews)**
- **Key fields:**
  - user\_id, business\_id, text, date
  - compliment\_count

#### **Use in RL:**

- Optional text-based feedback signal
- Can complement reviews for sentiment analysis or explanation

---

### **Schema Relationships**

User (user\_id)



Review (user\_id, business\_id, stars, date)



Business (business\_id, categories, attributes)



Check-in (business\_id, date)

---

### **Key Fields Mapped into RL Format**

#### **state: 5 recent interactions before action**

Each item in state includes:

- business\_id: ID of the visited business
- stars: Rating given (used for behavior modeling)
- date + time\_segment: Timestamp + time of day classification (morning/afternoon/evening)
- user\_profile: Dict with review\_count, average\_stars, friend\_count
- business\_checkin: Dict of check-in counts in each time segment
- business\_categories: Up to 5 most relevant business categories
- days\_since\_action + recency\_weight: Time-aware decay weighting (computed dynamically)

**action: The current business under recommendation**

Includes:

- business\_id
- action\_time\_segment: When the user acted on the recommendation
- user\_profile, business\_checkin, business\_categories: same format as state

**reward:**

- Set to 1 if stars  $\geq 4$ , else 0, as per DRL reward heuristics (Chen et al., 2023)

**next\_state:**

- Sliding window of the new state after action, formatted like state

**Output Files**

The processed data is saved to:

- rl\_dataset\_train.json – interactions before the 80% timestamp split per user
- rl\_dataset\_test.json – interactions after the split

Each file is saved in chunked JSON format, supporting scalability and future sampling or batch loading.