

Project 4 Proposal

Restaurant Recommendations Based on Client Reviews

Catherine Magsino

Background

I love finding new places to eat, and my number one place to go for restaurant ideas is Yelp. However, whenever I'm in this situation, I find myself spending too much time reading through restaurant reviews to get a feel of other people's experiences. The goal of my project will be to perform topic modeling for positive and negative reviews, and to provide a recommendation system for other restaurants with similar reviews.

Business Case

This project will have benefits for both the client and the restaurant owner. On the client side, it will provide an easy summary of reviews that will speed up decision making and help discover similar restaurants with the ones the user already likes. On the business side, restaurant owners can get a summary of the reviews and see where they can continue doing well, where they can improve, and how they compare to other restaurants.

MVP

My MVP will analyze each user's review as having either positive or negative sentiment, and will then group together topics for each type of sentiment. After my MVP is complete, I'll then compare the results against other restaurants to recommend restaurants with similar reviews.

Data

Data is retrieved from the [Yelp Dataset Challenge](#). While this dataset consists of 6 json files, I will focus on using 2 of the files (business.json and review.json) and will convert them to .csv files. The business.json file consists of business data including location data, attributes, and categories. The review.json file contains full review text data including the user_id that wrote the review and the business_id the review is written for.

Methodology

After cleaning my text, I'll perform topic modeling using LDA, NMF, and LSA.

Fields provided with each JSON file:

business.json

```
{
  // string, 22 character unique string business id
  "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

  // string, the business's name
  "name": "Garaje",

  // string, the full address of the business
  "address": "475 3rd St",

  // string, the city
  "city": "San Francisco",

  // string, 2 character state code, if applicable
  "state": "CA",

  // string, the postal code
  "postal code": "94107",

  // float, latitude
  "latitude": 37.7817529521,

  // float, longitude
  "longitude": -122.39612197,

  // float, star rating, rounded to half-stars
  "stars": 4.5,

  // integer, number of reviews
  "review_count": 1198,

  // integer, 0 or 1 for closed or open, respectively
  "is_open": 1,

  // object, business attributes to values. note: some attribute values
  // might be objects
  "attributes": {
    "RestaurantsTakeOut": true,
    "BusinessParking": {
      "garage": false,
      "street": true,
      "validated": false,
      "lot": false,
      "valet": false
    }
  },
}
```

```

    // an array of strings of business categories
    "categories": [
    "Mexican",
    "Burgers",
    "Gastropubs"
    ],

    // an object of key day to value hours, hours are using a 24hr clock
    "hours": {
    "Monday": "10:00-21:00",
    "Tuesday": "10:00-21:00",
    "Friday": "10:00-21:00",
    "Wednesday": "10:00-21:00",
    "Thursday": "10:00-21:00",
    "Sunday": "11:00-18:00",
    "Saturday": "10:00-21:00"
    }
  }
}

```

review.json

```

{
  // string, 22 character unique review id
  "review_id": "zdSx_SD6obEhz9VrW9uAWA",

  // string, 22 character unique user id, maps to the user in user.json
  "user_id": "Ha3iJu77CxlrFm-vQRs_8g",

  // string, 22 character business id, maps to business in business.json
  "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

  // integer, star rating
  "stars": 4,

  // string, date formatted YYYY-MM-DD
  "date": "2016-03-09",

  // string, the review itself
  "text": "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks.",

  // integer, number of useful votes received
  "useful": 0,

  // integer, number of funny votes received
  "funny": 0,

  // integer, number of cool votes received
  "cool": 0
}

```