**CS6240**

**Parallel Data Processing in MapReduce**

**Fall 2013**

**Project Proposal**

**BY**

**Purushottam Jha**

**Rushil Patel**

**Tushar Bhandari**

**Nishant Agarwal**

**Which data set will you work with? Where is this data set or how will you obtain it?**

We will be working with Yelp Academic Data Set. It will be available at the following link:

<https://www.yelp.com/academic_dataset>

We have requested for the data set and are awaiting a response from Yelp Team. How ever we do have a sample dataset of Yelp for Business around Phoenix Area.

**Describe in about one paragraph the high-level properties of the data.**

The dataset is a single gzip-compressed file, composed of one json-object per line. Every object contains a 'type' field, which tells you whether it is a business, a user, or a review.

**Business Objects**

Business objects contain basic information about local businesses. The 'business\_id' field can be used with the Yelp API to fetch even more information for visualizations. The fields are as follows:

{

'type': 'business',

'business\_id': (a unique identifier for this business),

'name': (the full business name),

'neighborhoods': (a list of neighborhood names, might be empty),

'full\_address': (localized address),

'city': (city),

'state': (state),

'latitude': (latitude),

'longitude': (longitude),

'stars': (star rating, rounded to half-stars),

'review\_count': (review count),

'photo\_url': (photo url),

'categories': [(localized category names)]

'open': (is the business still open for business?),

'schools': (nearby universities),

'url': (yelp url)

}

For Eg:

{"business\_id": "rncjoVoEFUJGCUoC1JgnUA", "full\_address": "8466 W Peoria Ave\nSte 6\nPeoria, AZ 85345", "open": true, "categories": ["Accountants", "Professional Services", "Tax Services", "Financial Services"], "city": "Peoria", "review\_count": 3, "name": "Peoria Income Tax Service", "neighborhoods": [], "longitude": -112.241596, "state": "AZ", "stars": 5.0, "latitude": 33.581867000000003, "type": "business"}

**Review Objects**

Review objects contain the review text, the star rating, and information on votes Yelp users have cast on the review. Use user\_id to associate this review with others by the same user. Use business\_id to associate this review with others of the same business.

{

'type': 'review',

'business\_id': (the identifier of the reviewed business),

'user\_id': (the identifier of the authoring user),

'stars': (star rating, integer 1-5),

'text': (review text),

'date': (date, formatted like '2011-04-19'),

'votes': {

'useful': (count of useful votes),

'funny': (count of funny votes),

'cool': (count of cool votes)

}

}

For Eg:

{"votes": {"funny": 0, "useful": 5, "cool": 2}, "user\_id": "rLtl8ZkDX5vH5nAx9C3q5Q", "review\_id": "fWKvX83p0-ka4JS3dc6E5A", "stars": 5, "date": "2011-01-26", "text": "My wife took me here on my birthday for breakfast and it was excellent.  The weather was perfect which made sitting outside overlooking their grounds an absolute pleasure.  Our waitress was excellent and our food arrived quickly on the semi-busy Saturday morning.  It looked like the place fills up pretty quickly so the earlier you get here the better.\n\nDo yourself a favor and get their Bloody Mary.  It was phenomenal and simply the best I've ever had.  I'm pretty sure they only use ingredients from their garden and blend them fresh when you order it.  It was amazing.\n\nWhile EVERYTHING on the menu looks excellent, I had the white truffle scrambled eggs vegetable skillet and it was tasty and delicious.  It came with 2 pieces of their griddled bread with was amazing and it absolutely made the meal complete.  It was the best \"toast\" I've ever had.\n\nAnyway, I can't wait to go back!", "type": "review", "business\_id": "9yKzy9PApeiPPOUJEtnvkg"}

User Objects

User objects contain aggregate information about a single user across all of Yelp (including businesses and reviews not in this dataset).

{

'type': 'user',

'user\_id': (unique user identifier),

'name': (first name, last initial, like 'Matt J.'),

'review\_count': (review count),

'average\_stars': (floating point average, like 4.31),

'votes': {

'useful': (count of useful votes across all reviews),

'funny': (count of funny votes across all reviews),

'cool': (count of cool votes across all reviews)

}

}

For Eg:

{"votes": {"funny": 0, "useful": 7, "cool": 0}, "user\_id": "CR2y7yEm4X035ZMzrTtN9Q", "name": "Jim", "average\_stars": 5.0, "review\_count": 6, "type": "user"}

**What analysis do you want to perform on the data? Describe the main analysis goals and any major sub-tasks that might be needed.**