**RMatcher: Recommendation Engine on the Yelp Dataset**

Ameen Askar - Joon-Sub Chung

**Introduction:**

We live in an age where we are overwhelmed with choices. This makes choice making a very difficult task, especially if one does not know what they are *exactly* looking for. With the advancement of search engines and huge databases, choice making has become slightly easier since the user narrow down choices, but it requires the user to know what they are specifically looking for and the search engine can still return choices that are not aligned with the user’s taste. We will be developing RMatcher, a recommendation engine, which sifts through the data users must search through, and display choices that the user would most appreciate. Our goal is to develop RMatcher so that it will lessen the requirements of finding specifics and allow the users to enjoy the results provided. The dataset we will be using has several user reviews of businesses and restaurants which have been provided by Yelp in their Phoenix, AZ metropolitan area.

**The problem:**

Given users’ ratings and reviews and other business related parameters like categories and times of most checkins, RMatcher will output a set of recommendations for each user. The list of recommendation will be based on the simple assumption that there are often many users who have similar tastes. RMatcher must be reliable with high precision and recall.

**Proposed Solution:**

We believe that our recommendation engine can be built with relevant results that meet the needs of the user. First, we will select an initial set of user ratings as a test set, and our RMatcher’s performance will be measured on the basis of precision, recall, and validity of recommendations provided to the user. We expect that the recommendation engine will return other businesses that have not been rated by the user and these results will be dropped in the automated analysis and reviewed manually randomly. After the test set is identified, we will need to create a heuristic for calculating the relevance of a recommendation to a user. We will be investigating different approaches to determine similarities, and we will have to test what model work best (User to user vs. Business to Business). Since we have non-binary data we plan initially to implement the algorithm based on Pearson correlation, which is very similar to cosine similarity. In addition to the Pearson correlation, we will include additional parameters that will, potentially, increase the model accuracy. For example, user weighting, reviews text sentiment analysis, and checkins data may contribute to the model sensitivity. We believe that by adding more parameters to the model, the recommendations will be more relevant.

**Milestones:**

4/26 Parse Yelp JSON data into relational DB - Identify a testing set - Research.

5/3 Basic pearson correlation Implementation - Sentiment scoring for reviews.

5/10 Implement our recommendation engine based on Pearson correlation of ratings,

sentiment scoring, categories, and checkins data.

5/17 Compare results to baseline method, and optimize weights for higher accuracy.

5/24 Cleanup code and results output - Final Report & Presentation.