The Yelp Helpers
Cynthia Tso, Ian Cross, Naman Choudhary
IMT 547

# Latent Semantic Analysis to Determine Similarity in a Corpus of Yelp User Reviews

**Abstract:**

The goal of this term paper is to assess a collection of Yelp reviews to see if reviews offered by Yelp Elite users are repetitive and not credible. Given that Yelp Elite users are nominated yearly (and can be self-nominated), is there a opportunity to game the system. By using a Natural Language Processing (NLP) approach to derive a "similarity scoring system", our team was able to assess reviews left by Elite Yelp users and regular Yelp users and notice a statistically significant difference in average similarity score between these two groups in a representative sample.

## Introduction:

How much does the general public really understand about Yelp's social ecosystem concerning frequent reviewers? The reviews-based social media site was founded in 2004 as an evolution of email listservs and the yellow pages, and has since evolved into the de facto go-to for online business reviews in 30+ countries.[1]

The company's revenue stream depends on an ad model, but that is not the focus of this research project. Paid advertisers (businesses) have no ability to change reviews, and Yelp has been very transparent about that process. Instead, we are interested in understanding how the social ecosystem that surrounds individuals who belong to Yelp's "Elite Squad", as Yelp does surface Elite reviews at the top of the review feed for a business.

According to Yelp's support center, the "Yelp Elite Squad" is a recognition bestowed by Yelp upon active members of the Yelp community, based on number of high quality reviews, tips, a thorough profile, and a "history of playing well with others".[2] The Elite Squad badge system functions similarly to the status offered by verification methods on other platforms such as Instagram, Facebook, and Twitter.

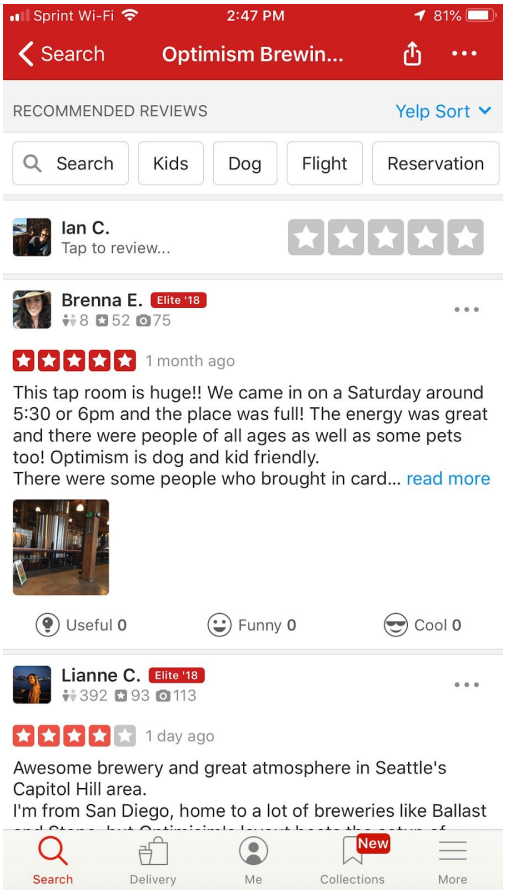


***Figure 1: Top Reviews for a Brewery***

[1] Yelp Factsheet - https://www.yelp.com/factsheet
[2] Yelp Support - https://www.yelp-support.com/article/What-is-Yelps-Elite-Squad?l=en_US

Interestingly enough, unlike those other platforms, verification to the Elite Squad is awarded on an annual basis and users have to be nominated (but can nominate themselves) to be considered for the upcoming year. This process could create a potential conflict of interest for those looking to preserve their social status at the expense of providing forthright reviews and tips for businesses. ***Is it possible that Yelp Elite Users tend to post reviews that are very similar in nature in order to pad their statistics and maintain their Elite status year-over-year?***

**Previous Work:**

Students of social media analysis looking to dive deeper into sentiment analysis of social media platforms like Facebook and/or Twitter, have studied Yelp reviews in the past. Even though their motivation, as well as the research question(s) were different, they had pretty much the same outlook as we do in this moment. Most of them however, have analyzed customer reviews as a means to improve businesses.

An example of NLP analysis of Yelp restaurant reviews[3] demonstrates how to analyze restaurant reviews on Yelp via topic modeling to determine main topics for positive and negative reviews. The analysis assigned each review a topic weight, and information was averaged to assign topic weights for each restaurant.With all restaurants and their reviews translated into a topic space, direct comparison of positive and negative review topics was possible.
The author used Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA), and Non-negative Matrix Factorization (NMF), giving us a great reference for our analysis. The author's analysis included the following steps:

- Exploratory Data Analysis
- Tokenization: (Yelp reviews) were tokenized using TF-IDF and 1-grams, which gave the most consistent results and distinct topics.
- Topic Modeling: using Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA), and Non-negative Matrix Factorization (NMF)
- Mapping All Reviews to Topic Space: Map all reviews to topics using Non-negative Matrix Factorization (NMF), normalize topic weights to sum to 1 to increase interpretability, average the topic distributions for all reviews of a restaurant to map that restaurant to the topic space.
- Visualizing the Results

Students at the Center for Urban Science and Progress at NYU studied yelp reviews to identify restaurant features using sentiment analysis.[4] Their methods for extracting characteristics of different types of restaurants was pretty similar to this paper's final method and consisted of two parts.

1. First, a Support Vector Machine (SVM) model was applied to differentiate positive and negative words in reviews, and further to get a word score to understand how positive or how negative the words were.

[3] NLP analysis of Yelp restaurant reviews:
https://medium.com/@Vishwacorp/nlp-analysis-of-yelp-restaurant-reviews-30b3d0e424a6
[4] NYU: https://arxiv.org/ftp/arxiv/papers/1709/1709.08698.pdf

2. Then the impact of scores, either negative or positive from different words within reviews of different restaurant category was analyzed.

They applied two different feature selection method for the SVM: “bag-of-words”: the frequencies of various words appeared in each review and “tf-idf”: the term frequency– inverse document frequency statistic. The labels were “positive” or “negative” distinguished based on the value the rating.

To get an ‘overall polarity score’ for each word, the sentiment score of each word was multiplied by its average frequency among all reviews. The accuracy of the SVM classifier on their test dataset was 88.906% with lambda setting to 0.0003. For the other feature selection method ‘tf-idf’, none of the regularization parameter lead to an accuracy over 88%.

**Data**

For this study, we made use of the Yelp Academic Dataset, made available to the general public by Yelp. Yelp also encourages students and others interested to analyze their dataset and participate in the Yelp Academic Challenge[5] wherein one can propose their hypothesis and conduct a study on the Yelp dataset to get to a conclusion.

The data was acquired a series of JSON files, containing roughly six million Yelp reviews from 10 metropolitan areas, regarding roughly 190,000 businesses. Due to computation and time constraints, analyzing the entire dataset was not feasible, so for an initial proof of concept we focused on sampling.

An initial proof-of-concept sample was randomly generated and contained about 3,000 unique users with the minimum and max number of reviews for each user ranging from 1 to 39. In latter stages of the the project, our team generated random samples from the dataset based on more specific requirements. The samples we generated contained 1,000 (Sample 1) and 2,000 (Sample 2) unique users with at least 20 reviews, the sample is sorted by user ID such that the sample contains all the reviews by a certain number of unique users. For Sample 1 this sampling method returned approximately 51,000 reviews vs. roughly 108,000 reviews for Sample 2.

In addition to the text of the reviews, the users JSON file contained information descriptive information such as the users name, count of total reviews, a list of their Yelp friends’ IDs, and crucially, a list of years that the user had been granted the Yelp “Elite” status.

**Methods**

As noted in the ‘Data’ section, the users were subdivided into groups of elite and regular users for each sample. As identified in the ‘Previous Work’ section, latent semantic analysis (LSA) provided the foundation for a scoring system to apply to each user in the sample. The LSA model takes a tokenized

[5] Yelp Academic Challenge: https://www.yelp.com/dataset/challenge

document corpus and dictionary as inputs. So in order to prepare the review data, our team normalized the reviews and built a corpus out of all the reviews for each specific user. The steps to do so are were as follows:

1. A normalization process to remove stop words, punctuation and convert all words to lowercase
2. A lemmatization process to lemmatize all verbs to their English root
3. Converting the normalized and lemmatized words to vector, the proverbial "bag-of-words"

In this representation, the corpus of all the distinct words used by a user are represented by one vector where each vector element represents a question-answer pair, in the style of: "How many times does the word system appear in the document? Once."

The n-gram bag of words are represented by integer IDs. The function *doc2bow()* simply counts the number of occurrences of each distinct word, converts the word to its integer word id and returns the result as a sparse vector that looks something like this [(0,1), (1,1)] which could mean that the words IDed '0' and '1' occur one time in the the first review[6]. Note in the case of an LSA model, TF-IDF can be a common method for input the document corpus. However, in the case of this application, and attempting to decipher the presence of "templated" or similar reviews, TF-IDF would not be an ideal scoring for term frequency. In TF-IDF, a term that appears less frequently across the corpus will have its weight increased relative to more frequent terms in the document corpus. **That increase associated with TF-IDF would be detrimental to assessing the presence of a pattern of similar reviews for a user as it would drown out the presence of words in a theoretical template with less frequent words.**

With a vector-based input to the the LSA model, the model can then compare each review made by the user against every other one of their reviews and generates a score based on their similarity. One way of computing this score between the vectors is by employing cosine similarity, which is a standard measure in Vector Space Modeling. Cosine similarity returns a score between -1 and 1, with 1 indicating exact similarity. In terms of this particular application, an average score closer to 1 would indicate a Yelp user has left very similar reviews.

[6] Gensim Documentation: https://radimrehurek.com/gensim/tut1.html

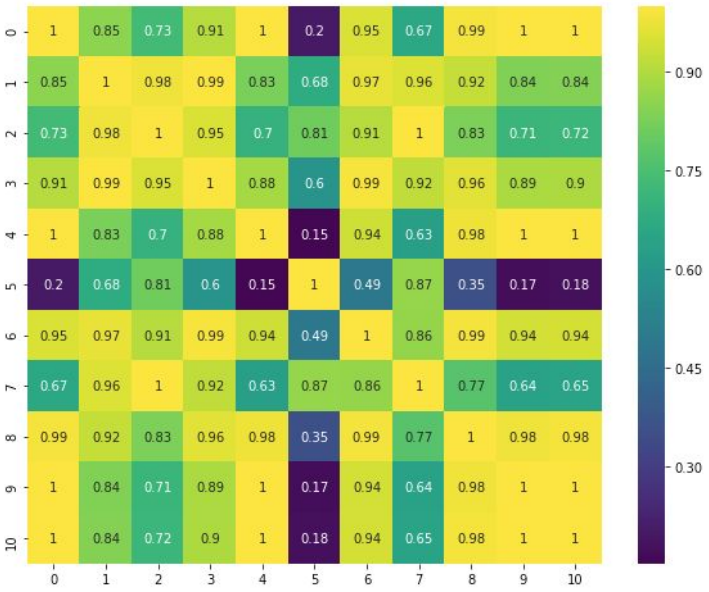


***Figure 2: Heatmap of Similarity Scores for Elite Squad Yelper "Gabi" in 2-D latent space (2 topics)***

There is some difficulty in aggregating a matrix of cosine similarities across groups of users. As demonstrated in the above figure, the model returns a cosine value between each document in the corpus. Most of the research in the field considers a single document corpus in its entirety, or endeavors to classify text in a certain way (i.e. positive/negative reviews). When comparing the scores in aggregate, do simply average each unique value under the diagonal of the matrix? Or is it better to set a threshold value for cosine similarity (i.e. .50) and count the number of document pairs that fall above/below that threshold? How also to assess the best number of topics to assess in latent space, when the scores start to converge? For our model, we ran the LSI model in the 'gensim' package available through Python at various topic levels.

Our team also transformed some of the *user* data to append the data frame to indicate in what years a user had Elite Squad status. This would be useful in applying further slicing of users in the next steps. For example, did Yelp Elite Squad users who had lost their Elite status produce reviews that had higher similarity scores in the subsequent years after losing the status?

## Results

As summarized in the density plot in Figures 3 and 4, average similarity scores across topic thresholds (ranging from 2 - 128) tend to be higher for the Elite users than regular users. When tested significance, each topic pairing of Elite vs. Regular users had significantly higher mean similarity scores than their regular counterparts as shown in the appendix. While it may be worthwhile to further assess the best fit for the model in terms of number of topics as an input, the fact that each user has their own corpus and dictionary and a diverse range of reviews (subject-wise) make it difficult to assess that fit in aggregate. For example you could run a cross-fold validation for each user, but that would make it difficult to compare across groups since each user would have a different number of topics. For additional clarity,

if this were a classification study (positive and negative reviews) like the previous work, it would make sense to set the number of topics at two for the model.

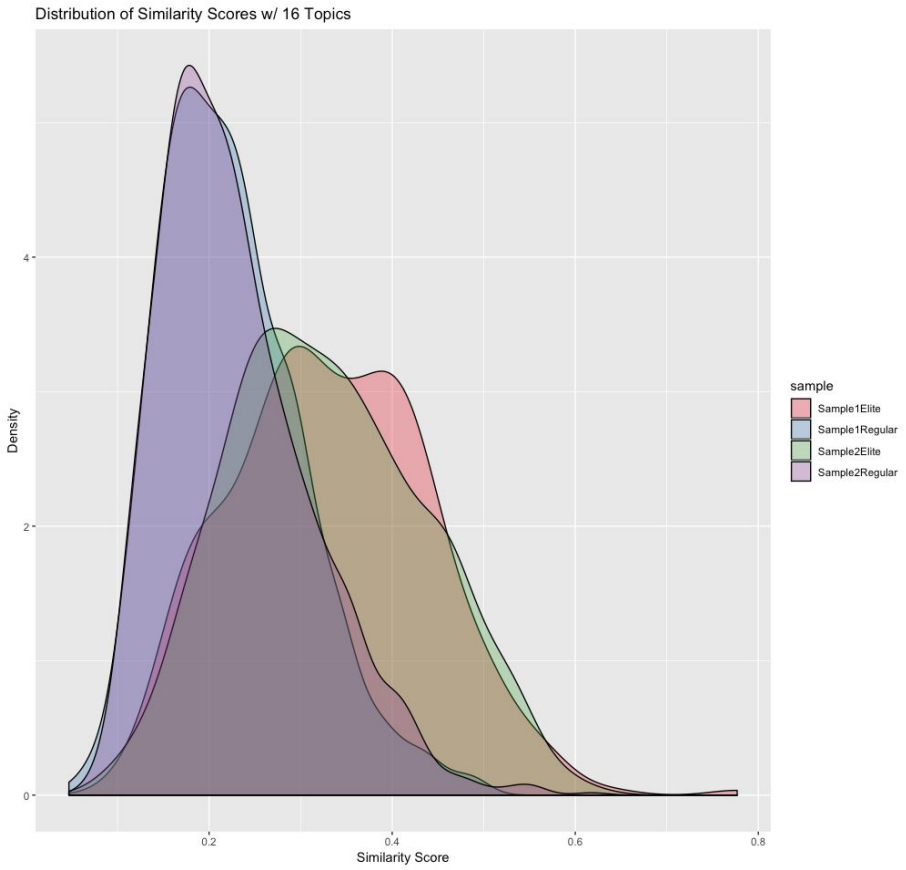


*Figure 3: Density of Similarity Scores across 16-D Latent Space*

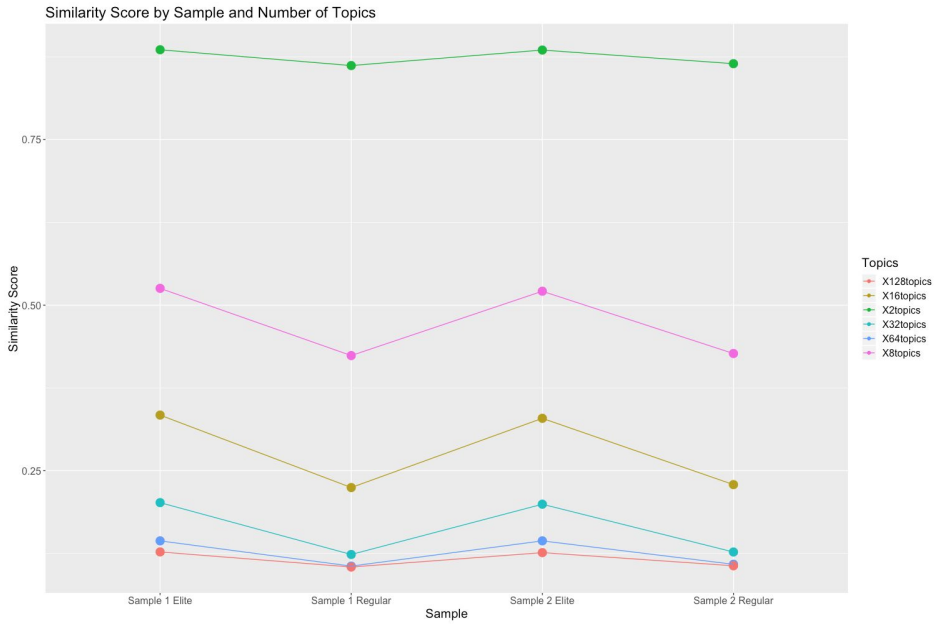


*Figure 4: Average Similarity Score by Sample and User Across Topic Thresholds*

Out team computed an alternative model that used TF-IDF/bigrams vector array to a Linear Support Vector Classification model to predict the authorship of a given review.[7] The model was able to predict authorship of Elite users with roughly 50% accuracy, and authorship of regular users with roughly 25% accuracy. However, predicting authorship does not directly answer the hypothesis that Elite users tend to post more similar reviews than regular Yelp users, and the number of reviews required by each user to improve the model accuracy would not have been useful in any practical context for Yelp.

## Discussion:

Even though our team was able to quantify the use of repetitive language and the general similarity between reviews, there are some significant caveats. Even using LSA and assigning values to specific words' contribution to the number of topics, the model would still struggle to assess context of a review. Also Yelp awards the 'Elite' status based on a user entire year-long corpus of reviews, across all business types. It would stand to reason though that the criteria for a review of a church or store might be significantly differ than a review for a restaurant and therefore might not templatable. So it may be worth paring down the sample to only include user reviews for restaurants. There could also be enhancements to the text normalization process. The method we used seemed to normalize best given the subject matter, but with additional time our team could have gone back to refine that or apply a minimum threshold for frequency.

There are other models that could have been useful in looking at similarity. Skip grams or continuous bag of words, wherein the model tries to infer the word surrounded by or surrounding the input words add a layer of context to the model, where word order and not just frequency matter. Those methods would really work to identify particular templates. In all realism, no analytical method would be able to adequately quantify similarity as human-beings have varying definitions what we consider "similar". Certainly it is obvious when there is word-for-word plagiarism, but anything less than that with semantic context clouding the issue, it would be difficult for two different human beings, let alone models, to look at the same review and agree that they are similar texts. There may be some research regarding plagiarism that could assist further study in this subject.

## Conclusion

While there are considerably more directions to take this analysis, the consistent findings of the average score at least indicate that our hypothesis was within the realm of possibility, and that Elite Yelp users do not provide a wide breadth of unique reviews. It can hardly be concluded from this analysis the most cynical scenario - that Yelp Elite users are actively templating their reviews to maintain their Elite status, yet perhaps the argument could be made that Yelp should surface more regular user reviews higher up on the fold when querying a restaurant (as exhibited in Figure 1). A special thanks to Professor Zeng for her patience during this project and assistance with our myriad questions.

[7] LinearSVC: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

## Appendix

*Table 1:*

| ***Sample 2*** | | | | | | |
|---|---|---|---|---|---|---|
| **Elite (N = 729)** | **2topics** | **8topics** | **16topics** | **32topics** | **64topics** | **128topics** |
| Min Score | 0.56 | 0.18 | 0.08 | 0.05 | 0.04 | 0.03 |
| Max Score | 0.98 | 0.75 | 0.63 | 0.50 | 0.50 | 0.50 |
| | | | | | | |
| **Regular (N = 1,271)** | **2topics** | **8topics** | **16topics** | **32topics** | **64topics** | **128topics** |
| Min Score | 0.43 | 0.18 | 0.07 | 0.04 | 0.02 | 0.02 |
| Max Score | 0.96 | 0.80 | 0.62 | 0.45 | 0.32 | 0.31 |

*T-Tests, Sample 2:*

| t-Test: Two-Sample Assuming Equal Variances | **Elite** | **Regular** | **Elite** | **Regular** | **Elite** | **Regular** | **Elite** | **Regular** | **Elite** | **Regular** | **Elite** | **Regular** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *2topics* | *2topics* | *8topics* | *8topics* | *16topics* | *16topics* | *32topics* | *32topics* | *64topics* | *64topics* | *128topics* | *128topics* |
| Mean | 0.89 | 0.86 | 0.52 | 0.43 | 0.33 | 0.23 | 0.20 | 0.13 | 0.14 | 0.11 | 0.13 | 0.11 |
| Variance | 0.00230 | 0.00301 | 0.00946 | 0.00904 | 0.01121 | 0.00677 | 0.00748 | 0.00275 | 0.00335 | 0.00135 | 0.00210 | 0.00122 |
| Observations | 729 | 1271 | 729 | 1271 | 729 | 1271 | 729 | 1271 | 729 | 1271 | 729 | 1271 |
| Pooled Variance | 0.00275072 | | 0.00919243 | | 0.00838857 | | 0.00447761 | | 0.00208212 | | 0.0015412 | |
| Hypothesized Mean Difference | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | |
| df | 1998 | | 1998 | | 1998 | | 1998 | | 1998 | | 1998 | |
| t Stat | 8.44913028 | | 21.1192788 | | 23.5276031 | | 23.2182122 | | 16.7050187 | | 10.8917287 | |
| P(T<=t) one-tail | 2.7838E-17 | | 7.8796E-90 | | 1.512E-108 | | 4.469E-106 | | 4.8205E-59 | | 3.5346E-27 | |
| t Critical one-tail | 1.64561663 | | 1.64561663 | | 1.64561663 | | 1.64561663 | | 1.64561663 | | 1.64561663 | |
| P(T<=t) two-tail | 5.5675E-17 | | 1.5759E-89 | | 3.024E-108 | | 8.939E-106 | | 9.641E-59 | | 7.0691E-27 | |
| t Critical two-tail | 1.96115201 | | 1.96115201 | | 1.96115201 | | 1.96115201 | | 1.96115201 | | 1.96115201 | |

*Background Data:*
https://drive.google.com/drive/folders/1JBXlU6feKcxjrSJCUx0TZsFSwE6y2YXg?usp=sharing