670 IR - Project Briefing
Find "spam" store! - A new Yelp score system based on reviews' text and reviewer authority

Jiehua LIU	Longhua YOU	Zhijiao LIU
322004845	423000667	522004372
loevaljh@neo.tamu.edu	luoyangylh@tamu.edu	zhijiao.liu.imp@gmail.com

ABSTRACT

Do you believe the official scores in Yelp's system? Do you ever consider the situation that some stores may hire workers to increase their scores? Our project aims to find out these "spam" stores! We proposed a new scoring system for Yelp consisting of two parts, SVM Based Auto Rating (SBAR) and Finding Gourmet(FG). By SBAR, the impacts of extremely short or meaningless reviews are effectively eliminated. By FG, gourmets are standing out, help give a reasonable score. Our scoring system is able to find out the 'spam' stores, and reviews are re-ranked according to the owners' authorities.

Keywords

SVM, expert authority, spam

1. INTRODUCTION

Do you believe the official scores in Yelp's system? Do you ever consider the situation that some stores may hire workers to increase their scores?

In this case, customers cannot get founded evaluations of each store. We question the score system, which only uses customers' rating. Reviews' text contains lots of useful info related to the features that determine whether the store is good or not.

Thus, we want to use the review content and customers' relationship between each other to build a new score system to re-score each store in different categories. After each store has their new scores, we will compare them to the official scores and find those stores that have huge difference between its new score and its official score, i.e. those stores are more likely have spam reviews.

Last but not least, we want to change the review showing timeline. Now the Yelp system show the reviews according to the posting time. But the most helpful review may be buried by other latest reviews. We will combine the qualities of customers' reviews and customers' relationship to give them "review authority", i.e. foodies' reviews are more influential, and help the useful reviews rank up.

2. RELATED WORK

A lot of efforts have been invested in information retrieval from reviews' text. Potamias[1] studied that on average the first ratings that businesses receive usually overestimate them. For tag mining, Kushal Dave[2] developed a system to find review's tag and associated sentiment score with them. And Lee, Sung Eob[3] designed a system to add tags with a negative/positive sentiment to a review. However, there are little research on predicating a rating based on reviews' text only.

Also, an observation on Yelp's website is that reviews are ranked according to their timestamp. The most recent posted review is ranked first in the website. This is undesirable for users usually want the most informative reviews. Another observation is that stars(rating) given by a reviewer contributes equally to a user's perspective. However, intuitively, we want reviews by experts (foodies, in our project) stand out among the ones from non-experts. This two observation motivates us, in addition to the machine learning based rating, to adopt as an important part of our whole rating system the expert-rating system. Finding experts has been a hot research area. Haveliwala et.al. [4] introduces topic-sensitive Pagerank. Weng et al. [5] find experts through link analysis.

In sum, we hope to develop an automatic scoring system to generate a reasonable rating of a review for restaurants.

3. PROBLEM STATEMENT AND SOLUTION

3.1 Problem statement

We question the official scores provided by Yelp, so we want to build a new stores scoring system to re-score those stores in our dataset. Our data come from Yelp Challenge, http://www.yelp.com/dataset_challenge/, which includes 15,585 businesses, 111,561 business attributes, 11,434 check-in sets, 70,817 users, 151,516 edge social graph, 113,993 tips and 335,022 reviews in the greater Phoenix, AZ metropolitan area.

3.2 Solution

Process:

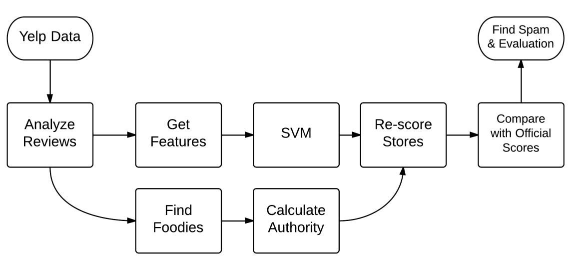


Figure 1 Project Process

Approaches:

- Finding foodies:

Find gourmet among reviewers.

Figure 2 Comparisons of the 2 Reviewers

Generally we'd like to prefer reviews of a foodie, one has professional knowledge of food and is able to give reasonable evaluations. Thus we hope to find foodies, or gourmets, among reviewers, give them priorities regarding scoring a restaurant, and rank top their reviews in the long list.

Feature Selection.

In Figure 2, we note some features that help a gourmet stand out. As in our project, we choose as features the number of Elite Award, the number of Total Reviews, the number of Votes, the number of Fans, and the number of Compliment. The more a reviewer gains, the more likely he/she is a gourmet.

Knowledge Propagation.

One common observation is that a friend of a gourmet is very likely to be a gourmet. Another is that a gourmet may prefer another platform other than Yelp to share his evaluation, thus may not be an active user in Yelp, however his reviews deserve much attention.

Based on these two observations, we build a network of friends, upon which the HITs algorithm is applied to calculate authority and hub score of each reviewer. Professional knowledge thus can propagate through this network, helping find 'inactive' gourmets.

Put Pieces Together:

A user score is assigned to each user. The higher the score one gains, the higher probability he/she is a gourmet. User score is a linear combination of the features, as well as hub and authority scores.

$$score(u) = \sum a_i \, b_i$$

The weights are tuned by experiments.

- SVM:

Feature selection

We selected 1000 restaurants, which are considered good rated, and about 30000 text reviews written about them from the experts we selected in the former part. To build the feature vector, first we picked the top K frequent words in all reviews, and this array of words will become our feature vector. Then we went through the reviews of each 1000 restaurants and counted the number of times that each word in the feature vector was used to describe that restaurant. At last, we calculated feature vectors for each restaurant by dividing the number of occurrence with the total number of occurrence of all words in the feature vector using this formula:

$$freq(i) = \frac{x_i}{\sum_{i=1}^{K} x_i}$$

where x is the number of times that i-th word in the feature vector appeared in the reviews of a restaurant. We chose K from 50, 100, 200, 300, and 400 to examine which one gives the least error. Given that we analyzed all the 3000 restaurants and calculated feature vector for each of them.

Model building

The feature vectors should be normalized or standardized in order to get a better classification. For example, if one feature is 100 times larger than another (on average), then the model may be better behaved if we normalize/standardize the two features to be approximately equivalent. So we normalized the feature vectors we got from feature select part to bring all of the variables into proportion with one another by using following formula to implement a unity-based normalization:

$$f_i = \frac{f_i - f_{min}}{f_{max} - f_{min}}$$

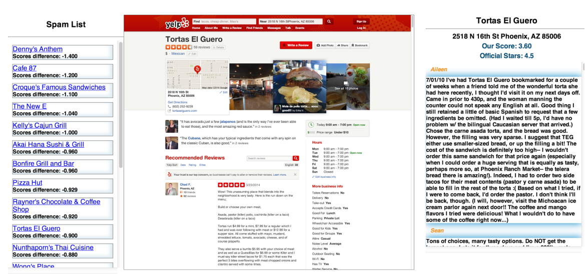
After feature normalization, we started to build the SVM model by dividing the data into 90% training and 10% testing dataset. For training data, we used both normalized feature vectors and official stars to train the model. For testing, we used our models to predict the business rating and then compared it with the actual rating that we had to evaluate the accuracy of our model. We use the Root Mean Square Error to quantify our error:

$$RMSE = \sqrt{\frac{1}{n} * \sum_{j=1}^{n} (y_j - \hat{y}_j)^2}$$

By calculating with different feature vectors, we found the best performance appears while having around 200 features, and RMSE is around 0.65.

Finally, we used the model we built to calculate business rating for all the restaurants in our dataset.

- Demo:

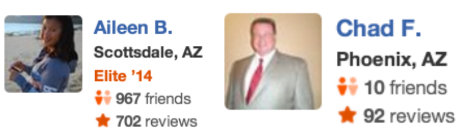


http://students.cse.tamu.edu/zliu4372/670_Project_FindSpamStores/

We build a web demo to show our results. It has three functions, as showing “spam” list, reranking reviews and comparing our system with original Yelp page. From the screenshot shown above, Top 20 “spam” stores are shown at left side according to the scores difference between our scores and the official scores. By clicking the name of the “spam” store, the information of this store and its reranked reviews will show at right side. The reviews in our system are ranked according to its reviewer’s authority, i.e. a reviewer with highest authority, his/her review of this store will be ranked top in our system. Meanwhile, the original Yelp page will change to the relevant store list based on the clicking activity. Thus, it is easy, useful and helpful to our users to compare not only the score differences but also the reviews’ quality.

4. EVALUATION

35.92% of the score differences in our system are lower than 0, which means around 1996 stores’ official scores are higher than our scores. 58.04% are lower than our scores. Only 6% official scores are equal to ours.



For example, the official score of Tortas El Guero is 4.5, but our score is 3.6. And after rerank the reviews, Aileen is more like a foodie than Chad, so in our system, her review is ranked top.

5. CONCLUSION

We proposed a new scoring system for Yelp, hoping that we can find out ‘Spam Stores’, which have unreasonable stars(score) given by the Yelp scoring system. Our system is consisting of two parts, SVM Based Auto Rating (SBAR) and Finding Gourmet(FG).

In SBAR, we collect representative reviews for some restaurants, extract features among these reviews, and train a SVM model. Using this SVM model we are then able to give each restaurant a score solely based on its reviews, which effectively eliminate the impact of extremely short or meaningless reviews.

In FG, we extract features from user profiles to help a gourmet stand out, giving him priorities and rank top his reviews. A friends network of the reviewer has also been built for knowledge propagation, in order to find ‘inactive’ gourmets in Yelp. Combining SBAR and FG, our scoring system is able to find ‘Spam Stores’, and re-rank reviews according its owner’s authority.

6. REFERENCES

[1] M. Potamias, “The warm-start bias of Yelp ratings,” arXiv e-print 1202.5713, Feb. 2012.

[2] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews,” in Proceedings of the 12th international conference on World Wide Web 2003, pp. 519–528.

[3] S. Lee, D. Son, and S. Han, “Qtag: tagging as a means of rating, opinion-expressing, sharing and visualizing,” presented at the SIGDOC’07: Proceedings of the 25th annual ACM international conference on Design of communication, 2007, pp. 189–195.

[4]Haveliwala, Taher H. "Topic-sensitive pagerank." Proceedings of the 11th international conference on World Wide Web. ACM, 2002.

[5]Weng, Jianshu, et al. "Twitterrank: finding topic-sensitive influential twitterers."Proceedings of the third ACM international conference on Web search and data mining. ACM, 2010.