# CS 229 Project : Improving on Yelp Reviews Using NLP and Bayesian Scoring

Patrick Bechon
pbechon@stanford.edu

Léo Grimaldi
leo.grimaldi@stanford.edu

Yacine Merouchi
merouchi@stanford.edu

## 1. INTRODUCTION

Yelp allows its users to share reviews of local businesses. For each business, the reviews and star ratings are used to display some key quotes from reviews, and an average star rating. Our goal in this project was to improve Yelp's user experience, by finding a new way of summarizing the ratings and reviews of each business. For our experiments, we used the Yelp Academic Dataset [1], which contains the data of the 250 closest businesses for 30 universities in the US. This dataset includes user profiles, business profiles, reviews, and the votes that users have given to other users' reviews. We used Bayesian scoring to improve the global star rating of every business, and then two methods, TF-IDF and ExpandRank (an algorithm derived from PageRank), to extract the keyphrases that best describe each business.

## 2. DATA PREPROCESSING

For Bayesian scoring, we only need the star ranking of every review, so, from the whole dataset, we extract a sparse 2D matrix whose element $i, j$ is the ranking of user $i$ for business $j$. One of the main characteristics of the Yelp dataset is that this matrix is very sparse, because many users only submit a low number of reviews.

For key-phrase extraction, we need to process the text of each review. To do so, we split the text into sentences and tokens. We remove all non-alphabetic words, we set all the characters to lowercase, we remove stopwords and we stem each word to regroup words with the same root. For all these operations, we use the nltk library in Python[2].

Among the different choices we had to make, the choice of the stemmer is probably the most important. We considered two algorithms, the Porter stemmer and the Lancaster stemmer. To choose between those two, we decided to test them on a supervised learning problem. The spam classifier of Problem Set 2 was our source of inspiration. By considering each review as a text and trying to predict whether it is a positive review (with a star ranking between 3 and 5) or a negative review (with a star ranking between 1 and 2), we were able to compare the performances of supervised learning algorithms using the two different stemming methods. Figure 1 shows the evolution of the training error and the testing error as the training set size increases. We chose to use the Porter stemmer as it outperforms the Lancaster stemmer on every test.

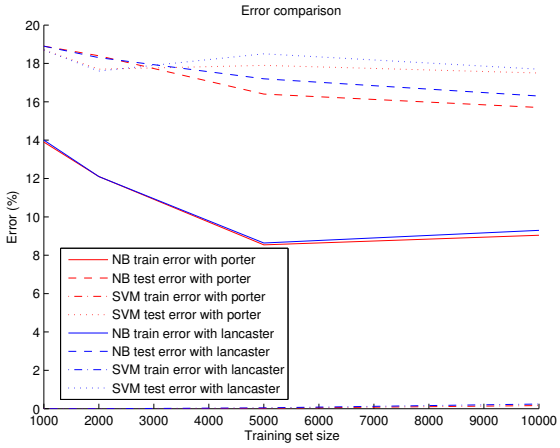


**Figure 1: Comparaison of training error and testing error on a 1000-reviews test set, for two stemmers, using SVM and Naive Bayes**

## 3. BAYESIAN RATING

### Introduction

On Yelp, reviewers can give a business an integer score between 1 and 5 ("star rating"). Then businesses can be ranked according to their average star rating among reviewers. Yet the variance within these average ratings is not very high and it is sometimes hard for users to pick their destination. Thus we want to come up with a better way of assessing the intrinsic value of a business $b$, denoted $\mu_b$, that would have a higher variance among businesses. While reviewers try to estimate this "true" value with star ratings, they can be inconsistent: while some tend to give all businesses a good score, others may be very harsh with their rating. Thus we denote $\nu_r$ the bias of reviewer $r$ and use a model similar to the one studied in Problem Set 4. The star rating given by reviewer $r$ to business $b$ is denoted $x^{(br)}$ and assumed to be

[1] http://www.yelp.com/academic_dataset
[2] http://www.nltk.org

generated by a random process as follows.

$$y^{(br)} \sim \mathcal{N}(\mu_b, \sigma_b^2)$$
$$z^{(br)} \sim \mathcal{N}(\nu_r, \tau_r^2)$$
$$x^{(br)} \mid y^{(br)}, z^{(br)} \sim \mathcal{N}(y^{(br)} + z^{(br)}, \sigma^2)$$

The variables $x^{(br)}$ (related to the business' true value and service consistency) and $z^{(br)}$ (related to the reviewer's bias and rating consistency) are assumed to be independent while the variables $x^{(br)}, y^{(br)}, z^{(br)}$) for different business-reviewer pairs are also jointly independent. We are to train this model on our dataset using an Expectation - Maximization (EM) algorithm.

## Regularization

Now the key difference with Problem Set 4 is that not all reviewers reviewed every business. We are even far from this ideal situation since each reviewer wrote less than 2 reviews on average. Thus for many reviewers, we don't have enough data to estimate the many parameters properly. While overfitting is a concern, the lack of data leads to a more crucial issue as the EM algorithm may fail to converge. Indeed, because of the very few data points available for some reviewers, some of the variance parameters $\sigma_b, \tau_r^2, \sigma$ tend to converge towards 0. These lead to degenerate Gaussian distributions (singular covariance matrix) hence singularities in the likelihood function. As the likelihood is no longer "smooth", the EM algorithm is not guaranteed to converge. In order to regularize the likelihood, we use MAP estimation with Inverse Gamma distributions as conjugate priors on the variances of our model's Gaussian distributions.

$$\sigma_b^2 \sim InvGamma(\alpha, \beta)$$
$$\nu_r^2 \sim InvGamma(\alpha, \beta)$$
$$\sigma^2 \sim InvGamma(\alpha, \beta)$$

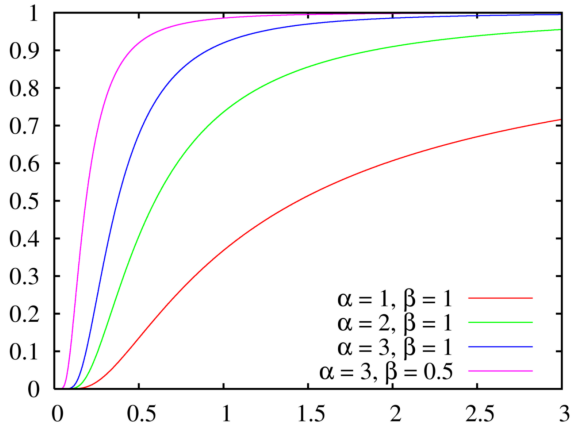


**Figure 2: CDF of Inverse Gamma Distribution for different values of $(\alpha, \beta)$**

As we can observe in the following expressions, the parameters $\alpha$ and $\beta$ can be simply interpreted as controlling the number of "virtual" prior samples that we add to the data in order to regularize the learning process. Thus $\alpha$ and $\beta$ must be high enough to prevent the variance parameters from converging towards 0, but low enough not to take over real data and pre-determine our model. In practice, we experimented on $\alpha$ and $\beta$, looking at the shape of the likelihood curve to determine small values that would make sure the algorithm converges over a finite number of iterations.

## EM Algorithm

E-step: For each business-reviewer pair $(b, r)$, we used the observed value $x^{(br)}$ and the current set of parameters $\{\mu_b, \nu_r, \sigma_b^2, \tau_r^2, \sigma\}$ to compute

$$\mu_{y,z|x}^{(br)} = (\sigma_b^2 + \tau_r^2 + \sigma^2)^{-1} \begin{bmatrix} \mu_b(\tau_r^2 + \sigma^2) + \sigma_b^2(x^{(br)} - \nu_r) \\ \nu_r(\sigma_b^2 + \sigma^2) + \tau_r^2(x^{(br)} - \mu_b) \end{bmatrix},$$
$$\Sigma_{y,z|x}^{(br)} = (\sigma_b^2 + \tau_r^2 + \sigma^2)^{-1} \begin{bmatrix} \sigma_b^2(\tau_r^2 + \sigma^2) & -\sigma_b^2\tau_r^2 \\ -\sigma_b^2\tau_r^2 & \tau_r^2(\sigma_b^2 + \sigma^2) \end{bmatrix}.$$

Thus,

$$Q_{br}(y^{(br)}, z^{(br)}) = p(y^{(br)}, z^{(br)} \mid x^{(br)}) \sim \mathcal{N}(\mu_{y,z|x}^{(br)}, \Sigma_{y,z|x}^{(br)}).$$

M-step: Denoting $R(b)$ the total number of reviews of business $b$, $B(r)$ the total number of businesses reviewed by reviewer $r$ and $N = \sum_b R(b) = \sum_r B(r)$ the total number of reviews, we used the quantities derived in the E-step to compute the updated values of the parameters:

$$\mu_b^* = \frac{1}{R(b)} \sum_r (\mu_{y,z|x}^{(br)})_1, \nu_r^* = \frac{1}{B(r)} \sum_b (\mu_{y,z|x}^{(br)})_2$$

$$(\sigma_b^*)^2 = \frac{2\beta + \sum_r \left(\mu_b - (\mu_{y,z|x}^{(br)})_1\right)^2 + \left(\Sigma_{y,z|x}^{(br)}\right)_{11}}{R(b) + 2(\alpha + 1)}$$

$$(\tau_r^*)^2 = \frac{2\beta + \sum_b \left(\nu_r - (\mu_{y,z|x}^{(br)})_2\right)^2 + \left(\Sigma_{y,z|x}^{(br)}\right)_{22}}{B(r) + 2(\alpha + 1)}$$

$$(\sigma^*)^2 = \frac{1}{N + 2(\alpha + 1)} \left(2\beta + \sum_{b,r} \left(x^{(br)} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \mu_{y,z|x}^{(br)}\right)^2 + \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \left(\sum_{b,r} \Sigma_{y,z|x}^{(br)}\right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right).$$

## Computation of the Likelihood

Using the distributions $Q_{br} \sim \mathcal{N}(\mu_{y,z|x}^{(br)}, \Sigma_{y,z|x}^{(br)})$ derived in the E-step, the lower bound

$$\phi(\underbrace{\mu_1 \dots \mu_B, \nu_1 \dots \nu_R, \sigma_1 \dots \sigma_B, \tau_1 \dots \tau_R, \sigma}_{\xi})$$

on the log-likelihood of the parameters, abbreviated as $\xi$, is given (up to an additive constant) by

$$\begin{aligned}\phi(\xi) = \quad & -\frac{1}{2\sigma^2}\sum_{b,r}\left(\left(x^{(br)} - \begin{bmatrix}1\\1\end{bmatrix}^T \mu_{y,z|x}^{(br)}\right)^2 + \begin{bmatrix}1\\1\end{bmatrix}^T \Sigma_{y,z|x}^{(br)} \begin{bmatrix}1\\1\end{bmatrix}\right) \\ & -\sum_{b,r}\frac{1}{2\sigma_b^2}\left(\left(\mu_b - (\mu_{y,z|x}^{(br)})_1\right)^2 + \left(\Sigma_{y,z|x}^{(br)}\right)_{11}\right) \\ & -\sum_{b,r}\frac{1}{2\tau_r^2}\left(\left(\nu_r - (\mu_{y,z|x}^{(br)})_2\right)^2 + \left(\Sigma_{y,z|x}^{(br)}\right)_{22}\right) \\ & -\frac{1}{2}N\log\sigma^2 - \frac{1}{2}\sum_b R(b)\log\sigma_b^2 - \frac{1}{2}\sum_r B(r)\log\tau_r^2 \\ & +\frac{1}{2}\sum_{b,r}\log\left(|\Sigma_{y,z|x}^{(br)}|\right) \\ & -(\alpha+1)\sum_b \log\sigma_b^2 - \sum_b \frac{\beta}{\sigma_b^2} - (\alpha+1)\sum_r \log\nu_r^2 \\ & -\sum_b \frac{\beta}{\nu_r^2} - (\alpha+1)\log\sigma^2 - \frac{\beta}{\sigma^2}\end{aligned}$$

This quantity was maximized in the M-step to update the parameters to $\xi^*$. But we can also use this expression right after the E-step to compute the likelihood of the current set of parameters. Indeed, we know that the distributions $Q_{br}$ computed in the E-step ensure that this lower-bound is tight for the current set of parameters $\xi$.

## Results

Since our dataset is organized around major US universities, we first trained our model on the businesses around Stanford. After determining the regularization parameters $(\alpha, \beta) = (5, 1)$, we experimented with three different random initializations for the EM algorithm and picked the model that gave the highest likelihood. First of all, people are

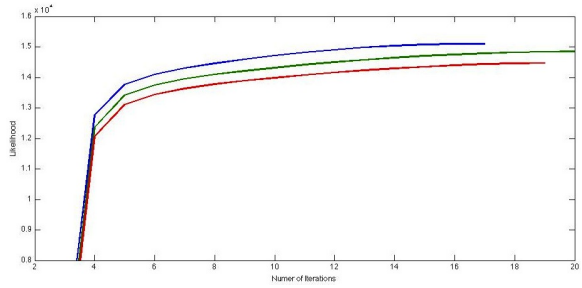


**Figure 3: Likelihood Convergence (Stanford)**

overly positive! Indeed the distribution of reviewers' biases range from -5 to + 5 and is packed around +1. Then we can observe that among Stanford businesses, the inferred "intrinsic" mean values $\mu_b$ have higher variance (2.21) than average star ratings (0.95). To the reader's appreciation, we also extracted the top 10 businesses according to each score (Bayesian Rating vs Average Star Ranking). Finally, we learnt a similar model for each of the other 29 universities from our dataset and compared people's mood (reviewer's average bias) and the quality of surrounding businesses (average business intrinsic value) around each university. It

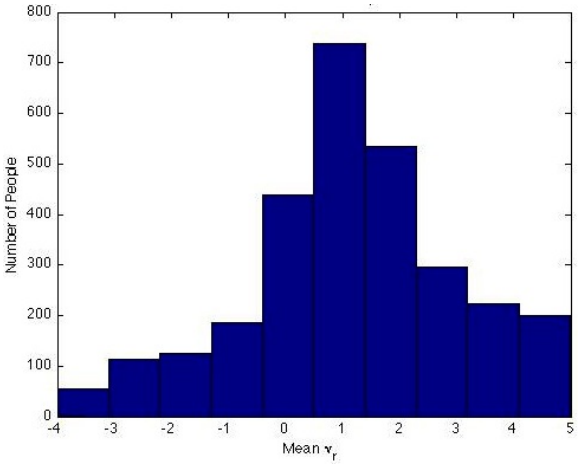


**Figure 4: Reviewer's Bias (Stanford)**

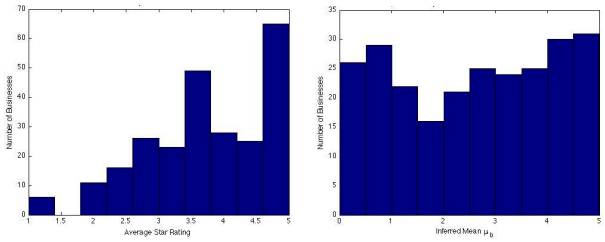


**Figure 5: Bayesian vs Average Star Ratings (Stanford)**

| Top 10 - Bayesian Rating | Top 10 - Average Star Rating |
|---|---|
| 1. CoHo Cafe | 1. Coleman Christie A MD |
| 2. Children's Health Council | 2. Sunken Diamond - Stanford Baseball |
| 3. Blood Center Stanford University | 3. Kc Esperance Np |
| 4. La Boulangerie | 4. Rodin Garden |
| 5. Stanford Medical Center | 5. Daniel Ponce, DDS |
| 6. Rodin Garden | 6. Gary Roberts, DDS |
| 7. Coleman Christie A MD | 7. Zaika Poi Dancer |
| 8. Welch Road Pediatric Medical Group | 8. Big Love Show |
| 9. Lacoste Boutique | 9. Cantor Rodin Sculpture Garden |
| 10. Stanford Taube Tennis Center | 10. Sam Most, MD |

**Figure 6: Top 10 Comparison (Stanford)**

looks like people are very positive here in California!

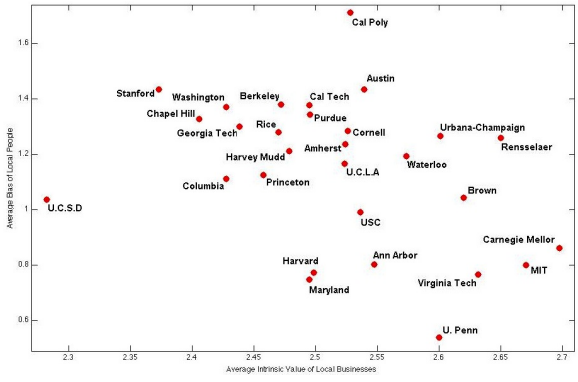


**Figure 7: Ranking of US Universities**

# 4. KEY-PHRASE EXTRACTION

## 4.1 Models

We consider the problem of extracting sentences that describe businesses. For this task, we use two popular methods available for extracting keyphrases, TF-IDF and ExpandRank. Both these methods involve assigning scores to words based on their importance, and then selecting sentences with the largest cumulative word scores. This task can be formalized as a unsupervised learning problem. We discuss these two algorithms and describe our evaluation of the results.

The idea of TF-IDF is to measure the importance of words relative to the documents in a corpus. We concatenate all the reviews from a given business and consider this as a document, and take only businesses from the same category for the corpus. We then calculate the TF-IDF score of each token relative to every business, and, for each business, we select the sentence with the highest cumulative word score. We eliminate sentences that are longer than seven words, to avoid a bias towards longer sentences when comparing cumulative scores and because we only want to display short sentences for the user.

The second approach that we consider is the ExpandRank algorithm, which was introduced in [1] using the example of extracting keyphrases from news articles. The ExpandRank algorithm works by selecting a small number of neighboring documents, assigning affinities to words by counting the number of times they occur within a fixed distance. This is similar to the PageRank algorithm, if we interpret every word as a web page and co-occurrence as the presence of a hyperlink. Instead of using a similarity distance as in [1], we use the business metadata to select neighboring documents, by restricting the businesses to a given category.

## 4.2 Evaluation of results

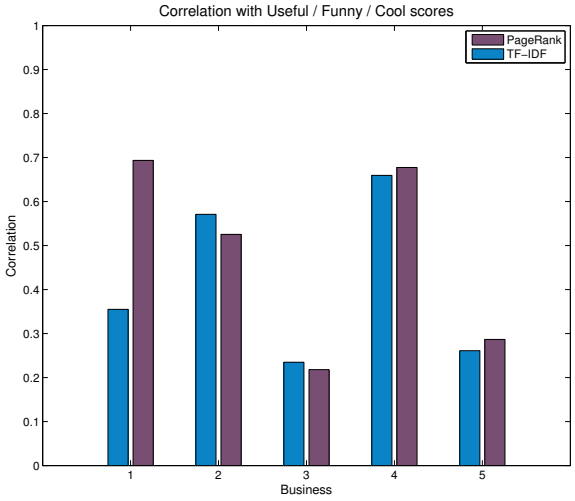


**Figure 8: Evaluation of key-phrase extraction algorithms using correlation with review votes**

Having no gold standard keyphrases at our disposal, we consider two methods to try to evaluate the results.

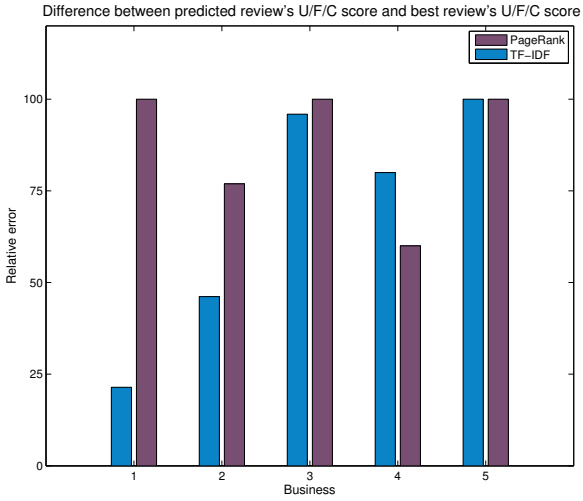


**Figure 9: Evaluation of key-phrase extraction algorithms using cumulative scores of reviews**

Our first idea is that the ExpandRank or TF-IDF score of a sentence might be correlated with the 'funny', 'useful' and 'cool' votes associated with the corresponding review. We compare the user's votes on reviews that we selected as containing the best keyphrase with the top review from the relevant business, and notice that there is no pattern. We then calculate the total cumulative scores of reviews using ExpandRank and TF-IDF, and find that these scores have a weak positive correlation with the user-submitted votes.

We also compare the results given by both algorithms on the same businesses. Given the same data, the two algorithms usually yield different results. We present an example of the results for five Mexican restaurants.

---

```
TF-IDF
1- We were at Lustre Pearl last night, ready for some tacos.
2- or me, the burrito ultimo is baja fresh's only saving grace.
3- Chef Jose Garces doesn't disappoint with Distrito.
4- took forever for our drinks to arrive.
5- Casa Moreno has the best house Margarita ever!
ExpandRank
1- might go back later at night for the food
2- Good fake mexican food.
3- Overall, good food, good drinks, and great ambiance/crowd.
4- Do not expect a good Mexican food experience if you go.
5- Do not order this dish unless you like spicy food.
```

In this example, as in other examples we have considered, ExpandRank seems to yield more relevant keyphrases. The word 'food' also appears in every sentence with ExpandRank, and we can verify that this word received a high individual score. This is due to the fact that this word is one of the most commonly used words in reviews of restaurants. The last example also shows a limitation of this algorithm: even the sentence is relevant, one needs more context to understand it (here the reviewer was talking about the chili verde).

# 5. CONCLUSION

This project proposes two ways of improving the user experience on Yelp: using Bayesian rating to adjust star ratings for the effects of user bias, and using keyphrase extraction techniques to describe businesses concisely. We found that the inferred intrinsic mean values can achieve a variance among businesses more than twice the variance of the average star ratings, and that we could deal with data sparsity providing a clever choice of prior on the model parameters. We compared TF-IDF and ExpandRank in the task of selecting keyphrases from reviews and found that they tend to provide different results, and only found a weak positive correlation with users' votes on relevant reviews.

Bayesian rating proved to be an interesting approach and future work could look at live implementation: how to adapt the algorithm for online update of the inferred grades at minimal cost, that is every time a person writes a new review. The results of ExpandRank are encouraging and this method should be tested on a large number of businesses with hand-selected keyphrases.

# 6. REFERENCES

[1] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pages 855–860. AAAI Press, 2008.