Sentiment Analysis on Yelp Reviews

We are Tuesday Group 3. Our team members are Siyu Wang, Xiaohan Wang, and Tiannan Huang.

Introduction and goal

Many people nowadays use Yelp to decide where to spend a lovely evening. Thus, Yelp is a place of many useful user-generated-content. We have two goals here. One is trying to predict the star of each review base on the review content. Second is trying to give some advice to the business owner by extracting information from the reviews.

Background information

We have four data here. review_train.json is the basic one, which has 5364626 rows and 4 columns (business_id, date, stars, text). review_test.json has 1M+ rows in the same format. business_train.json and business_test.json contain the business separately from the review_train.json and review_test.json. It contains some attributes and information about the merchants. For the business part, we use all the data. For the Kaggle part, we use 100k random sample.

Kaggle Prediction

We do this part by using two methods:

1. Bag of Words: Calculate the frequency of key words and use them as features
2. Deep learning Models: We keep the order of each word.

Bag of Words Model

1. PREPROCESSING

STOPWORDS: We basically use the built-in dictionary in nltk.corpus.stopwords, and we also add/delete some words.

GOOD / BAD WORD: Read in Pos & Neg dictionary by Minqing Hu and Bing Liu. We double the dictionary size by adding _NEG (Negative suffix) at the end of each word and put them into the reverse dictionary.

STEMMING: We use PorterStemmer() and WordNetLemmatizer()

TOKENIZING: Combine all the steps above, and write our tokenizer.

2. FEATURE SELECTION

We use CountVectorizer. Also, we use four methods:

1. Document Frequency (DF)
2. Chi-Square
3. Information Gain (IG)
4. Pointwise Mutual Information (PMI)

After comparing these four methods, IG is the fastest and the most accurate.

3. MODELING

We use three ways to form the attribute matrix:

1. Use their raw frequency
2. Use TF-IDF
3. Use Boolean version (Exist or not)

We use four traditional models:

- Naive Bayes
- Linear Model
- Logistic Regression
- SVM

The result on the test data are shown in the table below: (10k data, Train-Test-split is 0.8:0.2)

	Frequency-RMSE	TF-IDF RMSE	Boolean RMSE
Naïve Bayes	0.986	1.056	0.975
Linear Model	0.964	***0.892***	0.961
Logistic	1.101	0.933	0.953
SVM	1.000	0.909	0.959

Deep Learning Model

The structure we use is Input-Embedding-LSTM-Output. We use Databricks (Online Server) to run this part, there is a link to the report generated on that website. LSTM_Report (https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5112516171611286/1
Our final RMSE on the Kaggle test dataset is 0.81

Business Analysis

We want to give suggestions to a single merchant, thus we have to find a business which has the most reviews. We sort the business by the number of reviews and find that the top results are all in Las Vegas. In the table below, the dark blue denotes the restaurants on the strip.

			stars			
name	state	city	2.5	3.5	4	4.5
Bacchanal Buffet	NV	Las Vegas			8,568	
Wicked Spoon	NV	Las Vegas		6,887		
Hash House A Go Go	NV	Las Vegas			5,847	
Gordon Ramsay BurGR	NV	Las Vegas			5,575	
Earl of Sandwich	NV	Las Vegas				5,206
The Buffet	NV	Las Vegas		4,534		
The Cosmopolitan of Las V..	NV	Las Vegas			4,522	
The Buffet at Bellagio	NV	Las Vegas		4,318		
Luxor Hotel and Casino La..	NV	Las Vegas	4,240			
Lotus of Siam	NV	Las Vegas				4,131

From the table above, we decide to find out why Wicked Spoon in 3.5 stars while the other restaurants on the strip are 4 star and above.

Aspect Term Extraction (ATE)

We use attributes from reviews rather than business data. Because we want to give suggestions to a single merchant, thus the general pattern or trend of this field may not apply.

Our steps are as following:

1. We use spacy (Python package) to extract the root in a sentence along with its dependency.
2. Then we set up a dictionary to store the root and its dependency.
3. We sort the key in the dictionary by the frequency of the key itself plus the frequency of its dependency.
4. We manually go through the top results, select some candidates for further analysis.

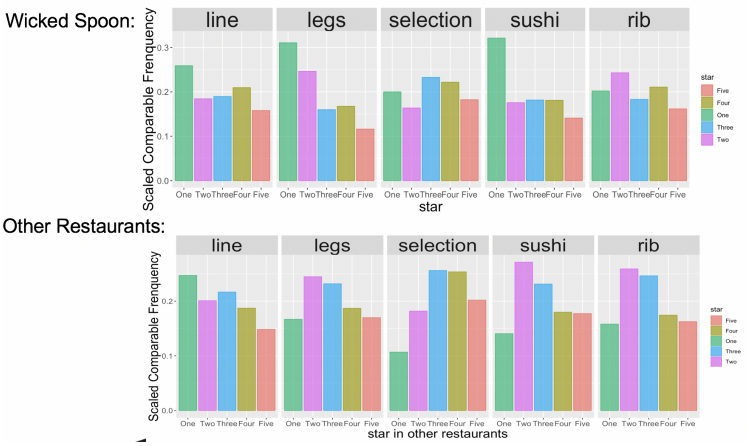
Below is a table of most frequent words in five-star reviews in Wicked Spoon

Word	Ratio of No. Mentioned	Some reviews
desserts	20.2%	a wide variety of **desserts** / look totally fresh and custom made / overall very pretty but unique looking.
cheese	16.8%	Devils **Mac and cheese**, soooo goooood!!!
marrow	13.3%	The **bone marrow** is a must try.
salad	10.7%	mini / beet / kale / caesar **salad**
chicken	13.0%	Fried **chicken** / the king pao chicken was exceptional.

Below is a table of most frequent words in one-star reviews in Wicked Spoon

Word	Ratio of No. Mentioned	Some reviews
line	28.8%	Ate at 6pm, waited in **line** to pay, waited in **line** to get seated then waited in **line** to serve myself.
legs	21.1%	Crab **legs** tasted like water / cold / frozen / broken / …
selection	18.9%	Limited **selection** / sucked / pitifull / joke / diminished
sushi	17.4%	there was no **sushi** worth mentioning / weird / hard
rib	12.8%	The prime **rib** was salty / fat / SO tough, I could not cut it with a knife.

Below is a distribution plot:



After we get the words, we do several hypothesis testing on them.

Hypothesis Testing

Because we use the star of the review as our response variable. It is a discrete variable, thus its distribution is not normal. (It means we can not use ANOVA / t-test)

For a specific word, the word divide the reviews into two categories:

1. The review contains this word.
2. The review does not contain this word.

Thus for these two groups, we do two tests:

1. Chi-square test: Test if two groups have the same distribution of stars
2. Wilcoxon two-sample test: Test if two groups have the same median stars.

The results are as follows:

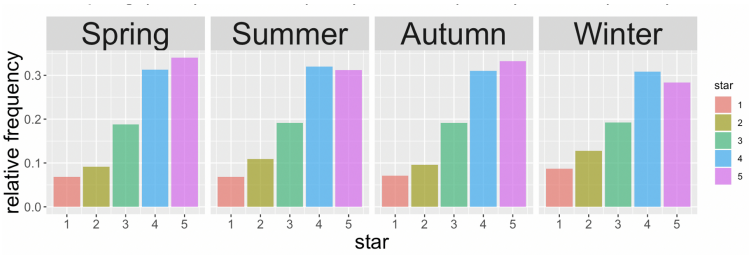
	Line	***Legs***	Selection	***Sushi***	Rib
P-value from Chi-Square	0.002	0.000	0.010	0.000	0.208
P-value from wilcoxon	0.003	0.000	0.182	0.000	0.046

Seasonal Trend

We use the month average score as our response variable and try to use time series analysis to gain some insights.

Pre-defined Season

We use four seasons, and we want to know if there is any difference between them. We do not have the normal assumption, so we decide to use the Chi-square test and Kruskal-Wallis K-sample test. (We put ANOVA here just as a reference)

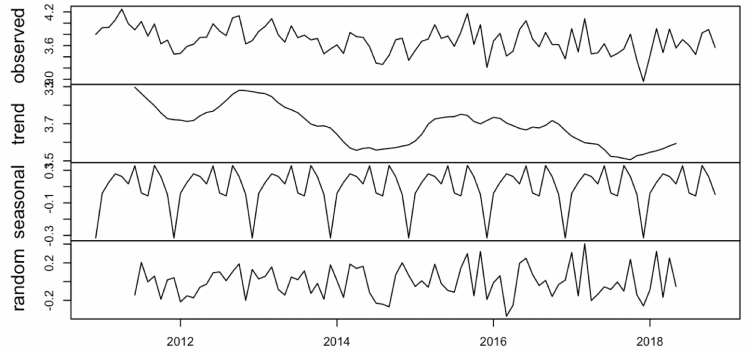


Hypothesis testing result:

	Chi-square Test	ANOVA	Kruskal-Wallis K-sample test
P-value	0.004	0.000	0.000

Seasonal Trend

We use MA model to decompose this time series:



We also use test the stationary of the random part. (p-value from the adf.test in r < 0.05 -> Stationary)

Then we decide to select some words from the reviews. We use the same method (frequency of word itself and its dependency) to select the most frequent words from reviews in December (df_dec) and other months (df_other). We select 100 words in both data sets. Then we select the symmetric difference between these two data sets.

In order to test if the selected word is important, we use two methods:

1. Frequency: eg. If the word 'line' appears a lot more in Dec than other months, it might mean that customers in Dec may face the problem of much more waiting time (need further verification).
2. Hypothesis testing.

The results are as following:

A/B	Percentage in december	Percentage other month	Wilcoxon_p	Chi-square_p	B/A	Percentage in december	Percentage other month	Wilcoxon_p	Chi-square_p
soup	1.50%	1.40%	0.29	0.444	spoon	3.90%	4.10%	0.31	0.502
reviews	6.10%	4.20%	0.13	0.464	staff	2%	3%	0.381	0.905
waiter	1.50%	1.60%	0.061	0.15	*potatoes*	**0.30%**	**1.70%**	0.871	1
meats	2%	2%	**0.048**	**0.002**	*flavors*	**0.80%**	**1.90%**	0.089	**0.005**
minutes	4.40%	2.70%	0.103	0.418	mimosas	1.80%	2.90%	0.359	0.671
crab	5.70%	6%	0.119	0.817	salmon	1.60%	1.80%	0.566	0.675
decor	2.30%	1.80%	0.338	0.469	cream	1.80%	2.70%	0.429	0.716
cocktail	1.10%	1.10%	0.123	0.78	salads	1.50%	2%	0.418	0.203
hours	**4.80%**	**1.80%**	0.155	0.486	area	1.10%	1.80%	0.993	0.168
money	5.20%	3.50%	0.312	0.953	friends	3.40%	3.40%	0.156	0.365
sauce	1.10%	1.50%	0.411	1	way	4.10%	4.70%	0.902	0.152

We select four words in the table above and go back to the original reviews to get some insights.

Conclusion

ATE:

- Line: Many restaurant on the strip are facing the same problem. So the review with the word 'line' mentioned is negative with high probability. If the merchant could add more tables in the restaurant and hire more cashiers at the front desk may be much help.
- Crab Legs: Many people complain that the crab legs are cold/frozen. So maybe change the menu (grilled or boiled) might be some help.
- Sushi: It seems that the restaurant on the strip all face the same problem. One way to solve this is just deleting the sushi from the selection. The second way is to hire some skilled cook and try to make sushi one of the special.

Seasonal Trend

- Meat: Some people say that the meat has no flavor while others say it is over salted. I guess in winter the business owner may have to make more effort on the quality control and try to keep consistency on how much salt to put.
- Hours: The potential pattern here is in a normal season, there is a shorter line in Wicked Spoon than other buffets, thus some people will choose to give Wicked Spoon a try. However, in winter (travel season), the line outside the Wicked Spoon is as long as other restaurants. Thus I think the business owner has to figure out a way to give more room capacity in winter.

Duties

- Siyu Wang: Deep Learning Models, Text Preprocessing and Jupyter Notebook
- Xiaohan Wang: Business Analysis part and Jupyter Notebook
- Tiannan Huang: Traditional Models and Jupyter Notebook

Reference

[1] Sentiment Analysis by Monkeylearn (https://monkeylearn.com/sentiment-analysis/)

[2] Databricks Keras Official Tutorial (https://docs.databricks.com/applications/deep-learning/single-node-training/keras.html)

[3] LSTM Model using Keras by Francois Chollet (https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html)