Topic Modeling as a Trend-Aware Performance Metric

Kenta Takatsu
Cornell University
Computer Science
kt426@cornell.edu

Caroline Chang
Cornell University
Computer Science
cdc222@cornell.edu

ABSTRACT

Past research has successfully predicted star ratings of Yelp reviews using Topic models, but has not used such star predictions to assess performance trends in specific businesses on Yelp. With this in mind, it is imperative to determine whether the topics found in the reviews can help us establish an accurate timely metric for users as well as business owners. We hypothesize that topic models become even more informative features when presented in a chronologically-sorted sequential format. In this paper, we devise methods to infer trends in Yelp businesses performance from latent review topics using different classifiers. The contributions of this paper are as follows: 1. the development of text embeddings which preserve the business-specific features that correlate with timely performance, 2. the qualitative analysis to provide a list of topics to improve business performance, and lastly, 3. the validation of our hypothesis that topic models in sequence demonstrate high prediction accuracy of business performance.

1. INTRODUCTION

Business performance can be highly volatile and is inherently tied to the response of users to the service a particular business provides. On Yelp, the primary feedback that businesses receive from their clientele is through reviews and star ratings. Even with these star ratings, there is no feature on Yelp that allows users as well as business owners to assess the "trend" of business performance as a restaurant, company, etc. In other words, these star ratings do not perfectly reflect time-sensitive information of whether or not given businesses are undergoing successful or deteriorating periods of business performance. Past research on the Yelp dataset has successfully predicted star ratings after generating latent subtopics from review text using Latent Dirichlet Allocation–also known as LDA [2]. However, past researchers have neither thoroughly validated the accuracy of such star ratings nor made an overarching conclusion about timely performance in businesses in these different categories.

Our main objective is to develop a streamlined star rating prediction system that can ultimately give users a long-term idea whether particular businesses are undergoing positive and negative trends in their own performance with the help of important subtopics that are relevant to that particular business. For this reason, we concluded that simply taking the average of N most recent ratings is not sufficient to assess the business performance. Rather, we aim to study more robust underlying trend of business by investigating the topics in reviews. We achieve this by building a classifier, which takes a sequence of topics and predicts the "successful-ness" of given sequence as well as the prediction of future trend. Different size of sequences as well as accumulation of star ratings over specified time intervals will give us the means to properly identify "successful-ness" trends for any given business in a pre-chosen category.

We aim to determine which classifiers (in this study: support vector machines, XGBoost, and multi-layer neural net) have the greatest and most accurate prediction power.

First, the support vector machine (SVM) technique is an example of a flexible supervised learning classification method. It is effective in high dimensional spaces and remains effective even when the number of dimensions exceeds the number of samples. In its decision function, it uses support vectors and different types of kernel functions can be specified, a metric to which it is extremely sensitive.

Second, XGBoost is an ensemble technique that takes a form of tree structure; weak learners are trained sequentially to correct the errors from the previous models. XGBoost is known for its computational speed and ability to make an accurate prediction with fairly small data size since the model keeps training until no further improvements can be made.

Finally, Multi-layer neural net is a well-known deep learning architecture for the supervised learning, using the update technique called back propagation. Multi-layer neural network has an extreme benefit from its nonlinear activation and multi-layer structure to discover nonlinear manifolds in the dataset.

One of the main challenges for star prediction in this study lies in the variability in the amount of review and star rating data that is provided for us in the Yelp dataset. We categorize this as an issue of data-richness, which refers to time variance among the reviews of a particular business as well as the number of reviews per business. Due to this, we need to choose the business reviews from categories that contain a significant number of businesses with ample review data (200+ reviews per business). Additionally, to achieve the best prediction results possible, we must also ensure high quality of our generated latent subtopics to develop fine-tuned classifiers that are able to predict star ratings over consistent time intervals. Given that certain types of topics tend to appear frequently in businesses of the same category, we use LDA to generate topic distribution vectors (referred

Figure 1: Baseline

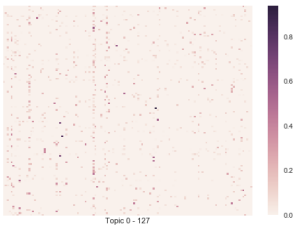


Figure 2: Augmented

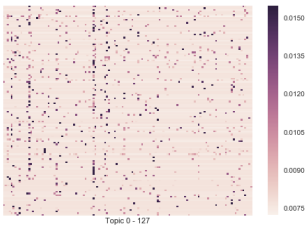
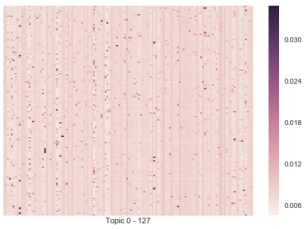


Figure 3: tfidf



The column of each heatmap above represents the latent topic, and the row represents randomly selected 128 reviews. Dense vertical line suggests the prevalence of the common latent topic (Figure 2 shows the frequent appearance of specific topics. In fact, the vertical line around the center contains words like "food", "great", "good", which do not provide any information). We can also observe how our Baseline embedding is prone to result low value in most reviews, while tfidf embedding shows randomness in data.

to in this study as embeddings) of business reviews from a given category where each element in a given vector corresponds to a topic distribution. In other words, we come up with embeddings that essentially translate a corpus of text into a vector space. Though our goal is to ultimately determine business performance using our developed classifiers, we also aim to devote a portion of this paper investigating whether types of topics themselves could also act as litmus tests of business performance, due to prevalence, changing popularity over time, etc.

The remainder of this paper proceeds as follows. Section 2 discusses the motivation and theory behind the three different methods for generating topic vectors: gensim, tf-idf, and word2vec. Section 3 details under what criteria we chose the Chinese restaurant category as a test case as well as the framework behind assessing topic embedding performance using correlation and predictive analysis. Section 4 discusses under what conditions and how our multiple classifiers performed in predicting star ratings as well as how well they were able to detect trends. Section 5 summarizes the content of this paper and alludes to future research.

2. EMBEDDING

Among the existing word embedding methods for natural languages such as word2vec [3], GloVe [5], we hypothesize that statistical topic models would capture more generalized trends that exists in latent space of review texts. In making such a decision, we deliberately disregard the sentiment of each word in corpus as well as specific sequence of them, but rather, we focus our study on the distribution of words in the corpus which represents the overall latent topics. In this paper, we use dimensionality reduction technique, Latent Dirichlet Allocation (LDA), and subsequently suggest embedding methods with a combination of LDA and tf-idf to better represent the characteristics of specific business. LDA has several advantages over other embeddings. Unlike the pre-trained vectorized models, LDA can be trained with smaller subsets of a large dataset. This gives us larger flexibility in tuning the scope of resulted topics. For example, we can run LDA over the dataset with any restaurant reviews and extract the general topics such as *Food*, *Good*, and *Service*. We can also generate more domain-specific topics such as *Bubble Tea*, *Cantonese*, and *Delivery* by filtering the original dataset so that we only examine the reviews for Chinese Restaurants. Additionally, using frequently explored methods like LDA allows diverse applications to specific situations. Dynamic topic modeling [1], optimized semantic coherence [4] are both modifying the baseline LDA to be adjustable to change in diction over time and topic coherence. In this paper, we are using online LDA as a base model; however, further studies can be done by comparing the performance among different LDA models.

In this paper, we propose three embedding methods that represent topic distribution of given review corpus: baseline embedding, augmented embedding and tf-idf embedding.

2.1 Baseline Embedding

The baseline embedding is a simple document-to-topic mapping with small modifications. Our initial exploratory analysis with documented-based LDA resulted poorly due to the fact that only one or two topics will get a value for each document (see Figure 1). Given a previous study in semantic identification in texts [6], we use their assumption such that the topic distributions can vary across sentences within a document. Mathematically, we define the baseline embedding as follows:

$$\theta_d = \frac{1}{n} \sum_{s \in d}^{n} Dirichlet(\alpha_s)$$

θ_d represents the embedded vector of document d, in this case a review text. s is a sentence in a document d, n is a number of sentences in d, and α is a parameter for Dirichlet distribution.

2.2 Augmented Embedding

The augmented embedding is a scaled method to assess the fact that most entries in the earlier embedding gets value of 0, which causes a vanishing gradient problem in classification tasks. This embedding also accounts for the fact that the baseline embedding gives implicit advantage towards a document with large number of sentences. Mathematically, we define the augmented embedding as follows:

Figure 4 Figure 5

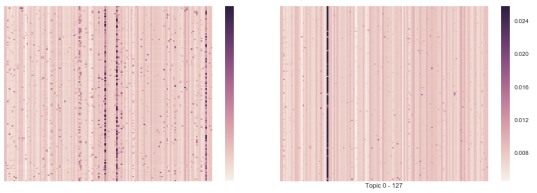


Figure 4 shows the tfidf topic embedding of random reviews from a 5-star sushi restaurant while Figure 5 is random reviews for a Chinese club. As we can see, even though they are random samples, the reviews for the same business contains similar pattern. The 2 vertical lines in Figure 4 are topics for "sushi" and "ice cream" while the strong vertical line in Figure 5 is a topic for "dance club"

$$\theta_d = \beta + \beta \times \frac{1}{n} \times \frac{\sum_{s \in d}^{n} Dirichlet(\alpha)}{max(Dirichlet(\alpha_{s'}) \text{for} s' \in d)}$$

β is a scaling measure to avoid zeroes in each topic. Also we are dividing the aggregated statistics from LDA by the maximum values in a document.

2.3 tf-idf Embedding

The tf-idf embedding is a further scaled embedding that represents importance of latent topics for each business. For this reason, tf-idf embedding takes two inputs: a review text and the business the review discusses. tf-idf, short for term-frequency-inverse-document-frequency, is a weighted measure that penalizes topics that appear frequently across all documents, and gives higher weights to small but meaningful topics. For example, sentences such as The food was good or The service was not nice might appear frequently but do not provide much information about the specific business. Instead, we want to give higher weights to information-rich sentences such as: Their traditional roasted duck was flavorful, which might not appear as frequently, hence the topic might disappear as part of noise without scaling. Mathematically, we define the tf-idf embedding as follows:

$$\theta_d(b) = augmented(d, \alpha, \beta) \times tfidf(b)$$

$$tfidf(b) = (\beta + \beta \times \frac{Dirichlet_b(\alpha)}{max(Dirichlet_b(_s) \text{for} s \in d_b)}) \times log \frac{N}{|d \in d_b : t \in d|}$$

Here, we are multiplying the result of augmented embedding by the tf-idf scaling factor, which takes all documents for given business as an input. In the tf-idf scaling, N represents the total number of documents in dataset (sentences in our case), and the denominator represents the number of times the each topic appears in reviews for given business. tf-idf embedding is a product of topic embedding and tf-idf scaling. We use the result from augmented embedding as a topic model, and tf-idf scaling is a metric to reward or penalize the topics based on how relevant each topic is to the given business. As we see in Figure 4 and Figure 5, this embedding was able to extract "most representative" topics of given business.

3. EXPERIMENT

In this paper, we conduct a series of experiments to measure both quantitative and qualitative performance of our topic embeddings as a business performance metric. As we hypothesize in the earlier section, the embeddings based on topic modeling should be able to preserve the implicit features that associated with business performance even though the model ignores the semantics and the sequence of each word in corpus. In order to quantitatively assess our embeddings as a business performance metric, we use a predictive task: construction of classifiers which predict business performance using our embeddings. The accuracy of such tasks implies the extent of which our embeddings can preserve the business performance in its latent space.

3.1 Dataset

In our study, we preprocessed the dataset so that it only contains the reviews for the *Chinese* category; this is to avoid the output topics from becoming overly generalized, such as *Food* and *Service*. We selected the *Chinese* tag for several reasons. Firstly, we observed the number of unique categories that co-occurred with *Chinese* tag. The underlying assumption is that number of co-occurred categories can be used as an indicator of the number of latent topics. *Chinese* tag appears with 135 unique categories among 1,240 categories the Yelp dataset contains; More than 10% of unique categories in the dataset co-appeared with *Chinese* tag, which suggests the large number of latent topics in *Chinese* business. Secondly, *Chinese* has higher average number of reviews for each business, compared to the other categories. The ample data for each business allows us to further focus our study on business-level. After preprocessing, the data contains 175,281 reviews from 3,773 businesses, written from December 2004 to July 2017.

3.2 Performance Metric

Initially, we used the Yelp star rating as an indicator of business performance; however, this metric did not represent the general performance of businesses very well. We hypothesized the reasons as follows:

- The rating is extremely skewed and sparse. There are ample 4-5 stars reviews, while not enough 1-2 stars. (see Figure 6)
- The sentiment of the rating is not universal across the businesses or the users. For example, a 3 star rating could both mean positive or negative based on the business and the user.

Given these observations, we define the business performance metric to be the difference between review rating and average business rating. After calculating this offset, we label the positive values as 1, negative values as -1, zeros as 0. The new offset metric can be interpreted as a positive, neutral and negative sentiment of reviews for particular business. We can also use this metric to define the business performance of sequence of reviews; in this case, we calculate the average of rating in the sequence of the reviews and find the offset from the average rating for the business. (see Figure 7)

3.3 Design

In order to generate the embedding, we first run LDA across our pre-processed data. Following the previous studies in sentiment analysis[6, 7], we chose the latent topic dimension to be 128 with 83678 vocabulary. Given this embedding, we conduct two different experiments to validate the performance of our topic embeddings as a business performance metric: correlation analysis, and predictive analysis.

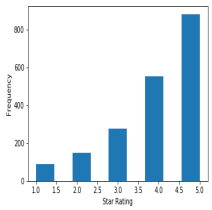
3.3.1 Correlation Analysis

In this experiment, we select 2 unique businesses with over 1,500 reviews and investigate the Pearson correlation between each latent topic and our performance offset. The Pearson correlation suggests the linear relationship between two continuous variables–in this case, the latent topics and the performance offset. Showing the strong correlation between the 2 values suggests highly interpretable association between topics and the business performance. We compare the results with different embeddings

Table 1: 5 Examples from LDA topics

Topic Name	Top 10 words
Dim Sum	*dim,sum,good,food,place,restaurant,service,chinese,dishes,cart*
Club Scene	*club, place, like, floor,great,night,hakkasan,people,dance,music*
Waiting Time	*table,food,came,minutes,seated,wait,service,waitress,ordered,got*
Happy Hour	*hour,happy,chips,good,drinks,salsa,place,tacos,food,great*
Dessert	*like,good,ice,cream,dessert,mango,sauce,delicious,sweet,ordered*
Bubble Tea	*tea,milk,boba,place,good,bubble,like,ice,drink,food*

Figure 6 Figure 7



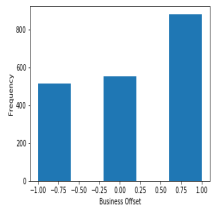


Figure 6 shows the distribution of Yelp star rating for sample data. We can observe the skewness in the data. Figure 7 is the same sample data but with the offset scaling. Three labels can be interpreted as positive, neutral, and negative sentiment of the review

and propose several interpretations. Furthermore, we define the vector of Pearson correlation as a characteristic vector, which identifies the importance of each latent topic to the particular business. Using this characteristic vector, we build a recommendation system by matrix factorization, which suggests most informative positive and negative feedbacks for each business. In matrix factorization, we represent users and items in a shared latent low-dimensional space of dimension K. The relevancy score is given by a following equation;

$$r_{ij} = u_i^T * v_j \tag{1}$$

In this equation, r represents the relevancy score of item j to user i. Usually, u_i is a vector representation of user feature and v_j is a vector representation of item features. We can use a correlation vector of length K (K = number of topics) as a user feature vector, the topic embedding texts as a v to recommend reviews (item) to business (user).

Using this concept, we calculated the score for each review and present the reviews with largest and smallest score. We validate the results qualitatively by observing the list of topics and reviews that most correlate with the success of business. This suggests highly interpretable application of our method as a business analytic tool.

3.3.2 *Predictive Analysis*

Before evaluating how well different classifiers perform using different embeddings (topic distributions) as features, we tested how effectively SVM in particular predicts star ratings on the *Chinese* review data as a whole using our baseline embedding. For every business in the *Chinese* category, we have run SVM on the businesss respective review data with a training sample > 50% and a testing sample, since we aim to have enough training data to get an idea of in what direction the business performance is headed. In this initial stage, the training and testing sets were randomly sampled in order to evaluate whether or not SVM would be able to accurately predict a star rating at a datapoint present at a particular time for a particular business. We conducted this preliminary experiment by randomly sampling a business out of a total of 3773 Chinese restaurant business ids, training at least half of the review data for that particular id, and assessing how far off the raw star ratings are from the predicted star ratings.

Offset Prediction

In this experiment, we validate the performance of each embedding quantitatively by giving a classification task. We construct multiple classifiers, which predicts the offset label of reviews given their topic embeddings. We use three classifiers: Support Vector Machine (SVM), XGBoost, and Multi-layer Neural Net. We compare prediction accuracy by k-fold stratified cross validation with k = 5. In order to control the variable, we repeat the same experiment with two datasets; the reviews for one business and the reviews for multiple business.

Sequential Performance Prediction

Due to the issue of data-richness, we will narrow down our sequential star rating prediction to categories that have an abundance of businesses containing more than 200 reviews, specifically the Chinese restaurant category. We will detail the method of training and testing our classifiers over a certain time interval and then subsequently predicting the next time interval bins star rating. We will then check the accuracy of that rating compared to original star rating and make an overarching assessment of prediction accuracies on all the businesses in the Chinese restaurant category.

4. RESULTS AND DISCUSSION

In this section, we present the empirical results from our experiments and our analysis of the results. In the experiment, we trained the LDA using gensim with 128 topics over 175,281 reviews from 3,773 businesses. The vocabulary is composed of 83678 words with all lower cases, excluding stopwords. From the output word distribution, we speculated the semantics of 5 subtopics and presented in the Table 1. As we can observe, the LDA was able to understand domain-specific concept such as *Dim Sum* and *Bubble Tea*.

4.1 Correlation Analysis

After the exploratory analysis, we decided to focus our studies on 2 particular businesses: SUSHISAMBA and Hakkasan Nightclub. We selected these businesses for the ample size of review data, and the distinctive trajectories in their business growth (See Figure 8, Figure 9). We embedded all reviews about these 2 businesses and calculated the correlation between each topic and the business offset – the difference between the review rating and the business average rating.

4.1.1 *Business Offsets*

We first compare the Pearson correlation between our three embeddings and the business offsets. Positively large correlation means that the topic represents important 'strength' of business characteristic. Similarly, we can speculate the negatively correlated topics as weakness of the business. As the Table 5 and Table 9 suggest, all embeddings output the same latent topics for being positively correlated with the business performance. We can qualitatively assess that this business is known for good *service*, *sushi* and *dessert*. Similarly for the different business, we can deduce that the positive traits are *club* and *fried food*. We can assess the negative traits of business in a similar manner. We also observe the recurrent topics across different businesses, which

Table 2: Recommended Reviews

Embedding	Most Negative	Most Positive
Baseline	I don't know where to begin... Let me just say I had the worst service by the rudest and most dismissive people at Sushi Samba - ever! (Rate : 1)	Love dining here every time I come to Vegas. The atmosphere is great and the food and drinks are worth coming.(Rate : 5)
Augmented	Probably the worst sushi restaurant we've been to. Dirty, loud, the service is terrible and the food is even worse.(Rate : 1)	Wow, what's totally cool Asian fusion sushi place! A Japanese and Spanish Peruvian mix.(Rate : 5)
tfidf	Not impressed with this Restaraunt. My fiance food came out late and cold. The waiter mixed up all of the bills.(Rate : 3)	This place has some unique flavors going on. To be totally fair we only tried the appetizers. The one that stood out in my mind was the duck confit tacos. (Rate : 3)

Figure 8

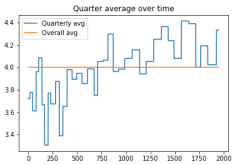


Figure 8 is a quarterly average for SUSHISAMBA. We can observe the growth in its business. On the other hands, Figure 9 shows the decline of Hakkasan Nightclub's business

Figure 9

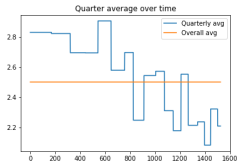
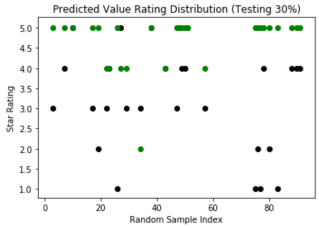


Figure 10



The majority of the predicted star ratings for this Chinese restaurant business is a 5-star rating. Green data points represent predicted star ratings whereas black data points represent raw star ratings. The calculated prediction accuracy for this business was 30%. We question where or not this metric is truly reliable, given that it consistently predicts moreless the same ratings each time, even with a 70%-train and 30%-test ratio.

is even particular in the negatively correlated topics such as *wait time*. This implies that the correlation analysis on negative topics failed to extract business-specific information, but rather, it is simply making an observation that people are prone to complain about serving time in Chinese restaurant reviews in general.

4.1.2 *Business Quarterly Offsets*

The results from the previous section did not capture the business-specific information, but rather, selected topics that strongly correlate with the offsets regardless of the business. This does not provide much information in terms of business analytic, in which we need to personalize the result between the business. Given this result, we investigated the correlation between the quarterly offset and the gradient of embedded topic over time. We naively implemented this task by binning the data quarterly (3-month intervals), and fit the linear regression. We then calculated the correlations between the coefficients of the linear regression and the offset of the quarterly average from the business average. The results are presented in the Table 13 and Table 17. In this observation, the positively correlated topics suggest the area associated with positive growth of the given business. Similarly, the negatively correlated topics are associated with the decline of the business performance. We can now observe much representative information about the business, demonstrating large variance across two samples. For example, we can assess how the *gluten free* menu increases the quarterly average of one business, and the *roast duck* for another.

4.1.3 *Review Recommendation*

The correlation analysis with the sequence of reviews shows promising results of our embeddings as a business performance metric. Following the equation (1) , we define the most informative positive and negative feedbacks for each business.

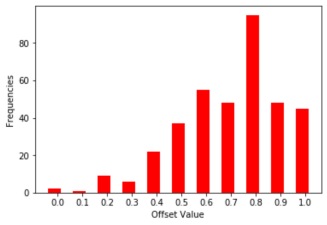
Baseline embedding resulted in recommending the reviews with extreme sentiment such as rating 1, and rating 5. The contents of reviews seem to capture rather general feedbacks; this is because the baseline embedding tends to become overly sensitive to the words like '*worst*' and '*best*', which appear in documents frequently. The result is presented in Table 2.

tfidf embedding, on the other hand, succeeds to capture more specific topics, such as name of menu, and the detailed experiences. Another interesting take-away is that both of most positive and negative reviews recommended by tfidf embedding is rating 3; however, the sentiment of them varies significantly. This suggests the sentiment of rating differ person to person, and our tfidf embedding was able to extract informative feedbacks based on contents, rather than the Yelp rating.

4.2 Predictive Analysis

In our exploratory analysis before the initial assessment of SVM on Yelp review data from the Chinese restaurant category, we have determined that the appropriate kernel for this particular classifier is radial basis function (RBF), a much more flexible parameter compared to the linear and multivariable kernels. The RBF kernel is known to act as a decent structural regularizer. By manual inspection, we determined that a 75% training data and 25% testing data allocation was appropriate, since with this proportion we would be able to generate predictions for a significant number of reviews while still aggregating most of the review star ratings as the training set.

Figure 11



For this sample of 900 businesses, the majority of prediction accuracy values occur at 80% with roughly the same number of business prediction accuracy values (60%, 70%, 90%, 100%). The x-axis represents prediction accuracy and the y-axis represents the number of businesses falling in every available prediction accuracy bin.

In Figure 10, we show an instance of running SVM on any given business that SVM will typically predict mostly the same one or two ratings across that businesss entire time on Yelp (from the time of the first review to the time of its most recent star rating). For any given business, we assessed how many predicted star values aligned exactly with their respective raw star rating, and used that percentage value to determine that particular businesss prediction accuracy. SVM had a consistent prediction accuracy of 33% across different businesses.

Furthermore, we took a random sample of 900 businesses and ran SVM with the same 70%-train and 30%-test distribution to determine the *degree* of accuracy with which SVM predicts, based on the trend it was trained for with 70% of the review star ratings for a particular business. We repeated this specific procedure multiple times since we have to traverse 3773 business ids in sections due to shear size of the JSON file. After we generate the predictions for a test sample from the raw star ratings of our test set, we generate an offset vector by subtracting the predicted star rating value from the raw star rating value for each of the star ratings in the test set. The assumptions we make in this study are that a good rating prediction is off by one star or an exact match. Any prediction that is more than one star off is a bad prediction. By this criteria, we determine the percentage of good and bad predictions within an offset vector belonging to a specific business (i.e. the number of good predictions / the number of total predictions or length of the offset vector). Figure 11 below shows a distribution depicting a more flexible prediction accuracy metric (due to integration of offsets as a margin of error) across a random sample of 900 businesses from the Chinese category with all the businesses with less than 10 reviews filtered out.

The typical range of prediction accuracy was between 0.5 and 0.9. Though this result indicates that SVM usually predicts at least half of the raw star prediction values for every business correctly, this coarse-grained prediction technique can be improved on.

Additionally, a significant finding in this portion of our study was that regardless of the number of features (topics) we selected to train on (from a single feature to 128 features), SVM still predicts the same star rating across the entire test set for a given business. We attribute this particular part of our result to the quality of our baseline embedding. In the set of 128 LDAgenerated embeddings, many of our topic distribution values are very close to 0, if not completely 0. This insufficient result confirms that we will have to explore different embeddings other than this baseline embedding in order to increase our prediction accuracy. The results of these are discussed as follows in the next section.

4.2.1 *Offset Prediction*

Out baseline SVM was not able to predict raw star rating. Given this result, we shifted the prediction from the raw star rating to out offset metric. Moreover, we label the offsets based on positive, zero, negative, and interpret them as positive,neutral and negative sentiment.

We conduct the k-fold stratified cross validation to assess the performance of classifier as well as the performance of topic embeddeings to measure the positivenegative sentiment of given review.

Embedding	Single Business	Mixed Business
Random	0.33	0.33
Baseline	0.5512	0.6615
Augmented	0.5589	0.5871
tfidf	0.5743	0.6103

Table 3: XGBoost Test Accuracy

Embedding	Single Business	Mixed Business
Random	0.33	0.33
Baseline	0.5205	0.5487
Augmented	0.4692	0.5513
tfidf	0.4615	0.5102

Table 4: Neural Net Test Accuracy

XGBoost

We set our XGBoost to have max depth as 6, and shrinkage parameter η as 0.3. In order to assess if the embedding can better represent business-specific information, we repeat the experiments with 2 datasets; the one with single business, SUSHISAMBA, and the one with mixed business. The Table 3 records the test accuracy of xgboost model. The model performing better at the mixed business suggests that the classifier was able to detect extremely positive and negative topics, such as *good* and *worst*; however the model is not particularly learning business-specific information.

Neural Network

We built a multi-layer neural network architecture with Tensorflow. We find the best hyper-parameters with grid search. The parameters are as follows;

- Number of hidden layers (1, 2, 3)
- Dimension of hidden layers (60%, 70%, 80%, 90% of previous layer)
- Dropout rate on the first layer
- The coefficient value β for the L1 regularlizer

After the grid search, we define the model with 2 hidden layers (the dimension of 100, 80 respectively), dropout rate of 0.1, and $\beta = 0.1$. The reason why we implemented dropout layer and β is, due to the small size of data, the model shows the tendency to overfit severely. In fact, we get 90% to 100% accuracy on the train data. We train the model for 50000 iterations and the result of cross validation is presented in the Table 4. Similarly to the xgboost, neural network failed to learn the business-specific information. Furthermore, due to its tendency to overfit, we concluded that the dataset is too small for neural network to converge without overfitting.

4.2.2 *Prediction of Time Interval Bins*

As we hypothesized earlier, topic embedding disregards word-based information such as order and semantics. For this nature, it is not surprising that the prediction of review semantics (positive, neutral, negative) only with the topic embeddings failed. Instead, we speculate that the prediction performance will increase if we bin the chronologically-sorted reviews together and use the sequence of the topics. As we define the gradient of topics

Figure 12: Baseline

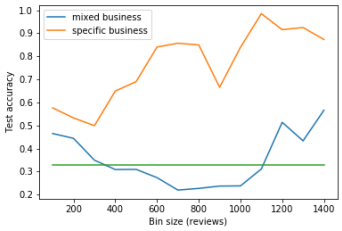


Figure 13: Augmented

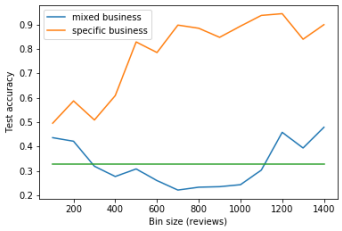
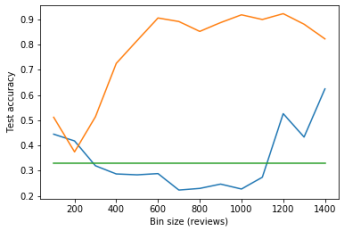


Figure 14: tfidf



The orange line is the test accuracy for the single business dataset, and the blue line is the test accuracy for the mixed dataset. For all embeddings, around 600, the accuracy marked 80%. With tfidf embedding, it achieved 90%. As you can observe, the sequential topic models show significantly higher prediction accuracy for single business dataset

by fitting the linear regression in Correlation Analysis, we use the same method to generate the sequential data.

We use the same xgboost model from the previous section, and generate data with different bin sizes. For each bin size i, we pick a review randomly from the dataset, select next i chronologically-sorted reviews and generate the sequential data (each dimension k' represents the slope of topic k).

We conducted k-fold stratified cross validation and plot the result in the Figure 12, Figure 13 and Figure 14. Likewise, in order to control the variable, we repeat the same experiment with single-business data and mixed-business data. In this graph, all emebeddings achieved over 80% accuracy once the bin size exceeds 600 while the prediction with mixed-business data stays low.

As we hypothesized earlier, the topic model seems to increase its prediction power when it's in a sequential format. Given this result, we propose the further studies to validate our hypothesis.

4.2.3 *Prediction of Future Performance*

As shown in Figure 10 as well as in Figure 15, for businesses with a huge number of reviews, it becomes difficult to depict an increasing or decreasing trend due to density of the data points on the plot. With a huge number of reviews for a business, 5 stratifications (one corresponding to ratings of each type of star) can typically be observed; subsequently, it becomes difficult to determine a trend from such highly. This difficulty can easily be alleviated by collapsing groups of data points into bins (e.g. a bin for every year) by taking the average of all the ratings in a given bin and plotting that single average point representing that group of ratings. Thus, the problem of dealing with a single-star predicting SVM might be resolved as well due to minimizing the chance of a densely populated number of certain types of star ratings that might bias the entire overarching prediction of a given business.

Instead of a strict 70%-train and 30%-test allocation among reviews for a particular business, we trained based on all the star rating data up until the third most recent years worth of star ratings. For determining most relevant business performance to users, we predicted the most recent 3 years of average star ratings. Figure 15 and Figure 16 show an example of predicting the average stars for the most recent 3 years a particular business has been active with the first embedding method.

We used the same criteria of good and bad prediction outcomes in this case. For offset vectors of a consistent length of 3, we consider a set of three predictions decent if differences between all the predicted star rating values and their respective raw star ratings are within 1. For the majority of businesses out of a random sample of 900 Chinese businesses , this was achieved. However, regardless of the three generated embeddings and the new binning method, SVM still predicts the exact same star rating value for the 3 most recently viewed years. A fundamental pitfall to this time interval binning experiment is that in this case SVM completely misses drastic upswings or downturns in performance trend over time in businesses. An example of this is shown in

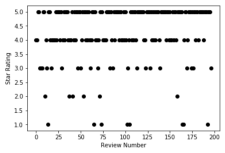


Figure 15: a

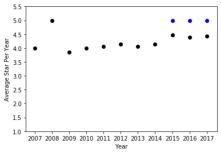


Figure 16: b

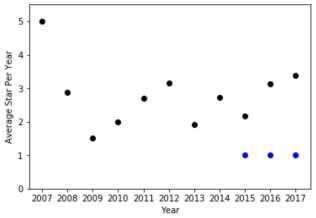
In Figure 15, we see an example of a 200-review business, showing a 5-layer stratification of 1,2,3,4, and 5 stars for a given business. It is not evident how to draw out a trend from this figure due to the lack of binning. In Figure 16, we have binned the review stars from business in Figure 15 by year and show SVMs prediction for the last 3 years–which maintains a stable trend as depicted in the original raw star values.

Figure 17, below. This puts into question whether or not we are effectively picking our training data the correct way. Further, this begs the question whether or not a different type of classifier should be used that would potentially predict ratings that are not always the same value.

5. CONCLUSION

Based on the prediction accuracy after implementing SVM classification, it is clear that for the majority of the time SVM has an over 50% prediction accuracy in star ratings for businesses in the Chinese restaurant category. However, we observed SVM to consistently return simply one (i.e. all 5s) or two types of rating regardless of time interval. We concluded that this is evidence of SVM incapable of predicting accurately when the test sets original star ratings show sudden opposite trend behavior. To improve the SVM classifier for the future, we will try different decision functions or even customize our own decision function instead of using RBF, since SVMs predictive power is very sensitive to the decision function. Multi-layer neural network could not combat the overfitting issue for this task; this is most likely due to the size of data, and overabundance of zeros in the latant topics which prevents the gradient to propagate backwards. XGBoost, on the other hands, demonstrated the remarkable prediction accuracy when the topic is presented in a sequence. This suggests the potential characteristics of topic models to provide valuable business-specific information when we study the sequence of them. Further study with advanced sequential models such as LSTMs should be done to validate our hypothesis. Regardless, the correlation between the gradient of latent topics and the average offset is already informative and can be utilized as a strong analytic tool for Yelp.

Figure 17



The average star prediction method using bins by year fails since SVM is unable to use the ultimate downward trend from 2007 to 2014 to predict the sudden positive review feedback from 2015 onward.

6. REFERENCES

[1] D. M. Blei and J. D. Lafferty. Dynamic topic models. *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 113–120, 2006.

[2] J. Huang, S. Rogers, and E. Joo. Improving Restaurants by Extracting Subtopics from Yelp Reviews. *iConference 2014 Berlin*, pages 1–5, 2014.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, pages 1–12, 2013.

[4] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, (2):262–272, 2011.

[5] J. Pennington, R. Socher, and C. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[6] F. Tian, B. Gao, D. He, and T.-Y. Liu. Sentence Level Recurrent Topic Model: Letting Topics Speak for Themselves. 2016.

[7] M. Yang, T. Cui, and W. Tu. Ordering-sensitive and Semantic-aware Topic Modeling. 2015.

Table 5: Correlation Table (Business Offset : SUSHISAMBA)

Positive topics		Negative topics	
Correlation	Topic words	Correlation	Topic words
0.3396	*food, great, service, place,friendly*	-0.2221	*food, like, place, rice,time*
0.1170	*like, good, ice, cream,dessert*	-0.2052	*order, food, said, minutes,ordered*
0.1148	*sushi, place, food, priced,vegas*	-0.1980	*food, service, place, bad, night*

Table 6: Baseline Embedding

Correlation	Topic words	Correlation	Topic words
0.3849	*food, great, service, place,friendly*	-0.2120	*food, like, place, rice,time*
0.1330	*like, good, ice, cream,dessert*	-0.2000	*food, service, place, bad,night*
0.1230	*sushi, place, food, priced,vegas*	-0.1976	*order, food, said, minutes,ordered*

Table 7: Augmented Embedding

Correlation	Topic words	Correlation	Topic words
0.3771	*food, great, service, place,friendly*	-0.2121	*food, like, place, rive,time*
0.1278	*like, good, ice, cream,dessert*	-0.2080	*food, service, place, bad,night*
0.1245	*sushi, place, food, priced,vegas*	-0.1967	*order, food, said, minutes,ordered*

Table 8: tfidf Embedding

Table 9: Correlation Table (Business Offsets : Hakkasan Nightclub)

Positive topics		Negative topics	
Correlation	Topic words	Correlation	Topic words
0.3154	*food, great, service, place,friendly*	-0.2660	*food, service, place, bad,night*
0.2635	*club, place, like, floor,great*	-0.2130	*time, food, card, good,service*
0.1164	*sushi, place, food, priced,vegas*	-0.1885	*order, food, said, minutes, ordered*

Table 10: Correlation table (Baseline Embedding)

Correlation	Topic words	Correlation	Topic words
0.2998	*food, great, service, place,friendly*	-0.2979	*food, service, place, bad,night*
0.1874	*tofu, good, food, sauce,dish*	-0.2385	*time, food, card, good,service*
0.1785	*dish, fried, chinese, good,food*	-0.1914	*order, food, said, minutes,ordered*

Table 11: Correlation table (Augmented Embedding)

Correlation	Topic words	Correlation	Topic words
0.2783	*food, great, service, place,friendly*	-0.2862	*food, service, place, bad,night*
0.2047	*dish, fried, chinese, good,food*	-0.2282	*time, food, card, good,service*
0.1780	*chicken, curry, rice, like,food*	-0.2149	*order, food, said, minutes,ordered*

Table 12: Correlation table (tfidf Embedding)

Table 13: Correlation Table (Quarterly Business Offsets : SUSHISAMBA)

Positive topics		Negative topics	
Correlation	Topic words	Correlation	Topic words
0.4000	*sushi, food, roll, buffet,time*	-0.5516	*table, food, came, minutes,seated*
0.3082	*food, good, gluten, free,rice*	-0.4973	*food, place, pho, good,hong*
0.2893	*place, food, dinner, good,chicken*	-0.4492	*food, good, time, chinese, chicken*

Table 14: Correlation table (Baseline Embedding)

Correlation	Topic words	Correlation	Topic words
0.5468	*food, chicken, ve, chinese,restaurant*	-0.4429	*table, food, came,minutes,seated*
0.4388	*food, good, gluten, free,rice*	-0.4353	*food, good, time, chinese,chicken*
0.4079	*kim, place, long, best,food*	-0.3727	*time, food, card, good,service*

Table 15: Correlation table (Augmented Embedding)

Correlation	Topic words	Correlation	Topic words
0.4608	*food, place, good, time,service*	-0.4688	*food, place, service, good,worth*
0.4183	*dim, sum, place, good,pork*	-0.4162	*good, food, place, service,restaurant*
0.3914	*food, good, gluten, free,rice*	-0.3968	*table, food, came, minutes,seated*

Table 16: Correlation table (tfidf Embedding)

Table 17: Correlation Table (Quarterly Business Offsets : Hakkasan Nightclub)

Positive topics		Negative topics	
Correlation	Topic words	Correlation	Topic words
0.6351	*food, delivery, ordered, chicken,order*	-0.6901	*good, like, service, restaurant,table*
0.5184	*food, pork, vegan, restaurant,service*	-0.4497	*food, place, good, time,service*
0.5174	*good, time, food, beef,like*	-0.4445	*food, nice, restaurant, time, ramen*

Table 18: Correlation table (Baseline Embedding)

Correlation	Topic words	Correlation	Topic words
0.5416	*good, time, food, beef,like*	-0.6662	*good, like, service,restaurant,table*
0.4545	*tofu, good, food, sauce,dish*	-0.5274	*food, restaurant, place, location,chinese*
0.4276	*food, chinese, place, good,service*	-0.5249	*food, chicken, rice, chinese,chino*

Table 19: Correlation table (Augmented Embedding)

Correlation	Topic words	Correlation	Topic words
0.5594	*duck, food, roast, chinese,place*	-0.6701	*good, like, service, restaurant,table*
0.5045	*good, time, food, beef,like*	-0.5995	*food, chicken, rice, chinese,chino*
0.5000	*food, curry, good, place,hotpot*	-0.4599	*chinese, food, good, pepper,ribs*

Table 20: Correlation table (tfidf Embedding)