Project Progress Report
Effect of Latent Topics in Online User Reviews and Tips on Restaurant Star Rating

Sahil Sanjay Gandhi, Aditya Kammardi Sathyanarayan

1. Changes

No changes have been made in the project scope or the methodologies with the exception of the size of the dataset. We originally mentioned that we would be running experiments on a subset of data, but during the data cleaning phase we realized that the dataset is manageable for iterative experimentations. Our approach is still the same (as originally defined in the proposal):

1. Extract latent topics using Topic Modelling and analyze distribution of topics per cluster (i.e. restaurant/geographic region) in the feature/dimension space
2. Assign per-topic rating per review and tip by either a simple average or a weighted average computation of the number of stars with the added context of the positive and negative weights of the neighbor words [1]
3. Predict the overall restaurant rating based on the extracted features by using Supervised Machine Learning techniques (Will also be used to evaluate extracted topics)
4. Analyze quantitatively, the effect of each latent topic on the overall star rating of the establishment. This will also be used to assert the "Volume vs Effect" for each topic, meaning we can find the topics that don't affect the overall star rating despite being in a lot of review text

2. Data Preprocessing

We're using the open Yelp Dataset for our project, in particular the reviews textual data and the tips textual data. It is evident from the proposal, that our approach relies heavily on Topic Modelling. Data cleaning and preprocessing is absolutely crucial for generating a useful topic model.

2.1. Data Selection

The Yelp reviews dataset is massive. The reviews dataset is about 4.5 GB alone with an addition of about 200 MB worth of tips textual data. The Yelp dataset contains various types of businesses, but since out project concentrates on restaurants, we filtered out all businesses that didn't contain 'restaurant' in the category type field. On further investigation, we realized that most of the reviews didn't reflect the trend displayed by other reviews. The review dataset contains a vote-count field called 'useful', which indicates how many users found the review useful. Most of the reviews that we thought didn't follow the review trend mentioned above had 0 'useful' votes. Hence, we decided to ignore all the reviews that weren't representative of the majority consensus by only considering the reviews that contained a non-zero 'useful' count. This resulted in filtering out more than 50% of the dataset.

2.2. Data Cleaning

In order to clean the data and remove irrelevant information, we followed the following techniques.

2.2.1. Tokenizing : converting a document to its atomic elements

Tokenization segments a document into its atomic elements [2]. In our case, we are interested in tokenizing to words. In our initial experiments we used NLTK's *tokenize.regexp* tokenizer.

2.2.2. Stopping : removing meaningless words

Certain parts of English speech, like conjunctions ("for", "or") or the word "the" are meaningless to a topic model. These terms are called stop words and were removed from the token list. However, the definition of a stop word is flexible, and the kind of documents (review and tip texts) may alter that definition [2]. For example, terms like "The Cheesecake Factory" will have trouble being surfaced because "the" is a common stop word and is usually removed. For the initial experiments we've decided to use simply remove the stop words without worrying about the problem explained above.

2.2.3. Stemming : merging words that are equivalent in meaning

Stemming words is another common NLP technique to reduce topically similar words to their root. For example, "stemming," "stemmer," "stemmed," all have similar meanings; stemming reduces those terms to "stem" [2]. This is very important for topic modelling, which would otherwise view those terms as separate entities and reduce their importance in the model. We use the Porter Stemming algorithm in our experiments.

2.3. Feature Generation

Based on the approach, we require 2 sets of features as input for the algorithms. First being the input to the Topic Modelling algorithm, and second being input to the Supervised Machine Learning algorithm. After data selection and cleaning, we converted the tokenized texts to a document-term matrix. This was fed into the topic model as input. The output of the topic modelling was the input for the supervised algorithm. Topic modelling generates n-topics (n being the number of topics) with a probability for each topic and the words in it. These topics per review were chosen to be the features for the supervised algorithm. Since for each review text, we have a star rating, we chose this to be the target values that the algorithm would predict.

3. Experiments and Results

3.1. Topic Modelling

After preparing the data based on the explanation above, we ran the Latent Dirichlet Allocation (LDA) algorithm on the document-term matrix. LDA is a topic model that generates topics based on word frequency from a set of documents. LDA is particularly useful for finding reasonably accurate mixtures of topics within a given document set. Here are some sample topics from our initial experiments:

Topic: Breakfast		Topic: Service		Topic: Thai cuisine	
Word	**Probability**	**Word**	**Probability**	**Word**	**Probability**
Egg	0.022	Food	0.016	Soup	0.017
Breakfast	0.015	Ask	0.011	Noodle	0.014
Food	0.014	Server	0.009	Pho	0.013
Toast	0.011	Tip	0.008	Roll	0.009
Waffle	0.010				

As you can see above, that the words are well-clustered to represent intuitive topics. The topic names were generated manually for representative purposes. Our research suggested we begin with a 50 topic LDA model and the above topics were generated from a 50-topic LDA.

The topic model provided us with vector representations for each review text across all topics (50 in our case). For our supervised algorithm, we followed the naïve approach by providing this review-topic matrix as the input feature matrix. The dataset contains a star-rating for each review, which served as the target values. By definition, the star-rating can lie in the range [0.0, 5.0], but in practice, we only found star-ratings to have the following values 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0. Since the target labels can also be seen as discrete classes, we ran both linear and logistic regression on this data. The preliminary accuracy score was at about 50%, but our main aim of running regression algorithms was to assess if we accurately determine a subset of the most important and least important topic categories across all review texts. This data backed analysis is extremely helpful for a company like Yelp as well as Restaurant owners.

4. What is Working and What is Wrong

In data preprocessing, we chose to use the regular expression tokenizer. This had side effects where words containing punctuation dropped characters after the punctuation point which resulted in half words that sometimes were not English language words at all. Similarly, the topic model generates some great distinct topics as shown in the table above. However, there were also cases where a single intuitive category was modeled as 2 or more separate topics in the output. In addition, we also found cases where the words didn't make sense to be grouped together. Analyzing the output, we also realized that the document-term matrix was a poor input choice for the LDA model. A TF-IDF matrix would've generated a better model.

5. Next Steps

Our project isn't complete, and we still have 2 more phases of the proposed approach to complete. But the immediate task is to improve the models built so far. We will replace the regex tokenizer with a more robust tokenizer, like the word tokenizer; Replace the document-term matrix with a TF-IDF matrix as the input for the LDA algorithm; Improve the stop words filtration to account for the relevant usages of stop words. Lastly, we'd like to experiment with kernels for the regressions algorithms to better model the data relevant to the dimension of the data.

6. References

[1] James Huang, Stephanie Rogers, Eunkwang Joo. "Improving Restaurants by Extracting Subtopics from Yelp Reviews" University of California, Berkeley. Link - https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_ImprovingRestaurants.pdf

[2] Link - https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html