Exploratory Analysis of Yelp Restaurant Reviews

Juan Luis Herrera Cortijo (juan.luis.herrera.cortijo@gmail.com)

1 Introduction

Today, customer reviews in social media have a deep impact on the chances of success of any business [1]. Restaurant customers look for a complete and satisfactory experience regarding food quality, service, and ambient and they often seek the opinion of patrons when they are choosing a place for their next meal. Learning which topics are the most frequent among customer reviews and how they associate to a positive or negative rating can help business improve their offer and have a better chance of succeeding.

In this report, I explore some latent topics in a corpus of Yelp reviews for restaurants.

2 Methods

2.1 The Data

The data used here is part of the Yelp Dataset Challenge.The dataset consists of a set of JSON files that include business information, reviews, tips (shorter reviews), user information and check-ins. Business objects list name, location, opening hours, category, average star rating, the number of reviews about the business and a series of attributes like noise level reservations policy, etc. Review objects list a star rating, the review text, the review date, and the number of votes that the review has received. This exploratory analysis focuses on this two type of objects.

I have filtered the business by category to keep only those businesses in the restaurant category (14303) and reviews related to those businesses (706646).

The texts from restaurant reviews will form the corpus for this analysis.

2.2 Corpus and language model

I have processed each of the reviews to build a bag of words language model. To create this model I preprocessed each document in the corpus as follows:

- Remove non-writable characters.
- Strip extra white spaces.
- Lower case.
- Remove punctuation
- Remove numbers
- Stemming
- Stop words removal.

After that, each text was tokenized into unigrams, and the unigram frequencies were counted and stored into a document-term matrix of counts.

Term counts across all the corpus showed a typical Zipf distribution. I kept the most frequent terms that, summing all their frequencies, accounted for about 99% of the total number of words in the corpus. The resulting vocabulary has 15697 words.

I used the R packages “tm” [3] (v 0.6) and “RWeka” [4] (v 0.4-24) for these steps.

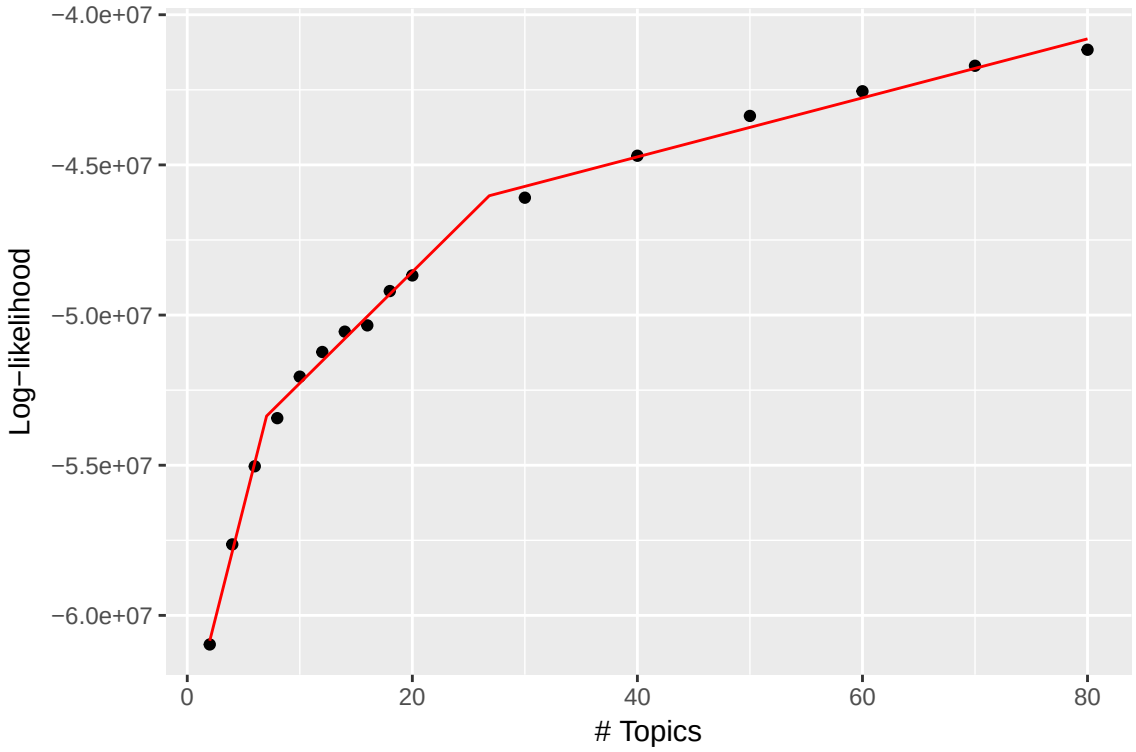

Figure 1: Figure 1. Topic model log-likelihood for a range of topic numbers. The red line is a result of fitting a three segment linear regression to the data.

2.3 Topic model

To discover latent themes in our corpus, I run a Latent Dirichlet Allocation [2] algorithm (LDA) using the document-term frequencies matrix as input. To estimate the model parameters we used a Gibbs [5][6] sampling with a burn-in phase of 1000 iterations and later the distribution was sampled every 100 iterations during 2000 iterations. I tested other approaches (LDA with VME parameter estimation and a Correlated Topics Model [7]) but the topics obtained were less clear than the ones resulting from LDA with Gibbs sampling.

To select the number of topics (k), I run LDA on 20% of the documents in the corpus (141298) using different k values. Figure 1 shows the log-likelihood for a range of values for k.

Finally, I decided to use 20 topics by fitting a three-segment linear regression and selecting the number of topics about the middle of the second segment. This method, similar to the elbow rule, seeks to get a simple model with enough flexibility.

```
## Warning: 'rbind_all' is deprecated.
## Use 'bind_rows()' instead.
## See help("Deprecated")
```

I used the R package "topicmodels" [8] (v 0.2) to compute the models in this report.

3 Results

First, I will examine the results of fitting a topics model to the whole restaurant reviews corpus. Second, I'll show the results of running the same analysis over two corpora, one for positive restaurant reviews and

another for negative restaurant reviews.

3.1 Overall topics

Most of the 20 topics obtained are well defined. Figure 2 shows them and the four most frequent words for each topic. Color luminance shows the relative in-topic relevance of each term, being the most bright the most common word. There is only one topic that I couldn't identify.

There are several topics about the customer experience: Disappointed, Love, Returning, Waiting, Good service, Nice, Take Away and having a Special Dinner.

The remaining topics focus on the type of restaurant, according to their cuisine type: American, Asian, Mexican, Italian, Sushi; or other features like Buffets, Location and Ambient.

There is a "No restaurant" topic because, although all the reviews are for businesses labeled as restaurants, some of them are not (for example, hotels). Also, the "Nightlife" topic focuses on clubs, music venues and other businesses related to night entertainment, but that are not restaurants.

3.2 Topics by rating

The overall topic model lists several topics about customer experience. But good and bad experiences are mixed because the corpus includes both positive and negative reviews. I have explored the topics related to positive and negative ratings independently. Figure 3 displays the restaurant review rating distribution. We can see that positive reviews (stars >3) dominate over negative reviews (stars <3).

```
## Warning in mutate_impl(.data, dots): Unequal factor levels: coercing to
## character
```

```
## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector
```

```
## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector
```

```
## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector
```

```
## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector
```

```
## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector
```

I have fitted two topic models (one for positive reviews and another for negative reviews) with 20 topics each and following the same methodology that I used to compute the overall topic model. I didn't include reviews with three stars because that rating is not positive nor negative. To avoid an unbalanced number of reviews in each category, I used a random sample for the positive category with size equal to the number of reviews in the negative category.

In general we can see some of the overall topics also appear in these two new models we get a finer grain topic distribution about the customer experience.

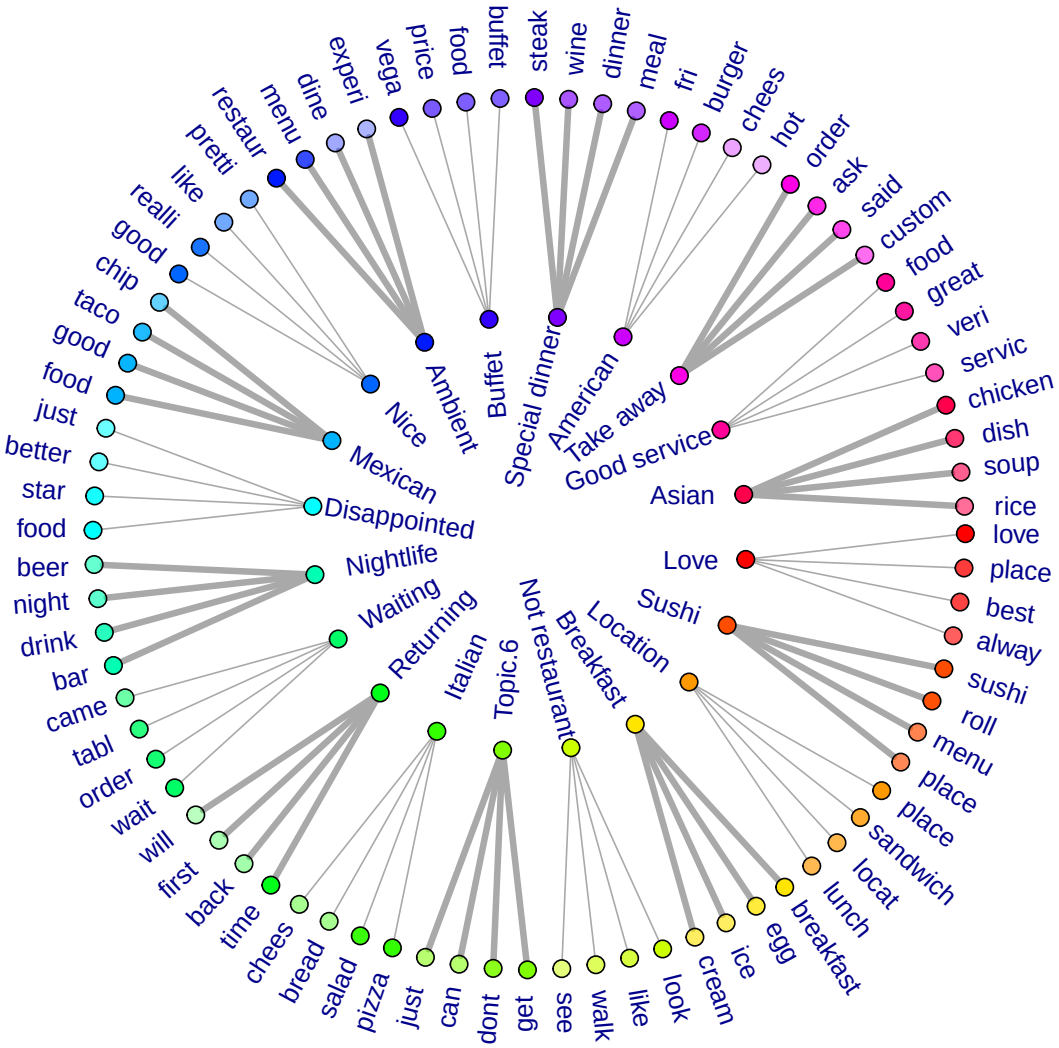

Figure 2: Figure 2. A topic model for Yelp restaurant reviews. Only the 4 most frequent words are shown for each topic. Color luminance shows the relative in-topic relevance of each term, being the most bright the most frequent term.

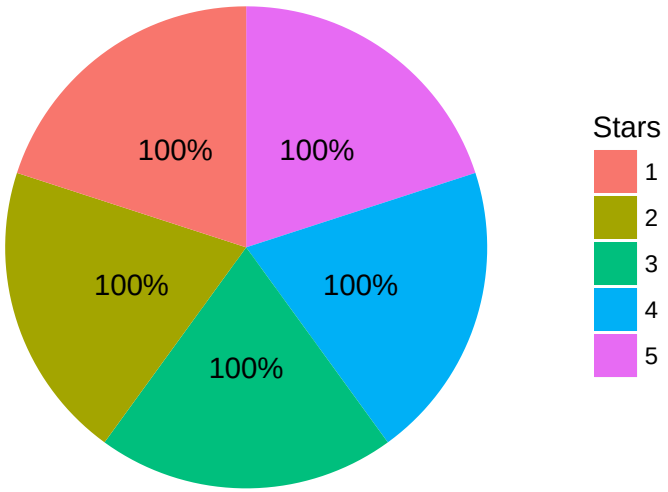

Figure 3: Figure 3. Restaurant review ratings distribution.

3.3 Topics by cuisine

Each business object includes a list of categories that we have already used to filter the reviews and keep only those reviews related to restaurants. In addition, business can be labeled with a certain type of cuisine: Italian, American, Shusi, etc. Since the overall topic model shows some topics related to cuisine types, now I'will investigate how topics are related to cuisine categories.

First, I examine how the topics found relate to the number of restaurants that belong to a certain cuisine. Figure 3 displays the relative relevance of each cuisine category in the business data. As we can see, the topics found match the most frequent cuisine types.

Now, I explore which topics are related to each cuisine. An exam of positive reviews shows that they display food descriptions more often than negative reviews. So, for this analysis I have used only positive reviews (stars > 3).

I have built a new document-term matrix in which each document represents a cuisine category and includes all the reviews related to business labeled with that category. For each word in the vocabulary and for each cuisine, I have added all the counts for that word in the rows of the overall document-term matrix that correspond to positive reviews for business with that cuisine label. This is equivalent to concatenating all the positive reviews for that cuisine and recomputing the document-term matrix.

I have fitted a topic model with 20 topics. Figure 7 shows the 9 most frequent terms for each topic. Now, most topics are related to some type of cuisine and just one is related to the customer experience (Love).

Figure 8 shows the probability distribution of the 25 most frequently reviewed cuisines over the topics in the model. Not surprisingly, all the cuisines show a high probability in the topic love because we are using only positive reviews. Also, the probability distribution is coherent with the cuisine topic.

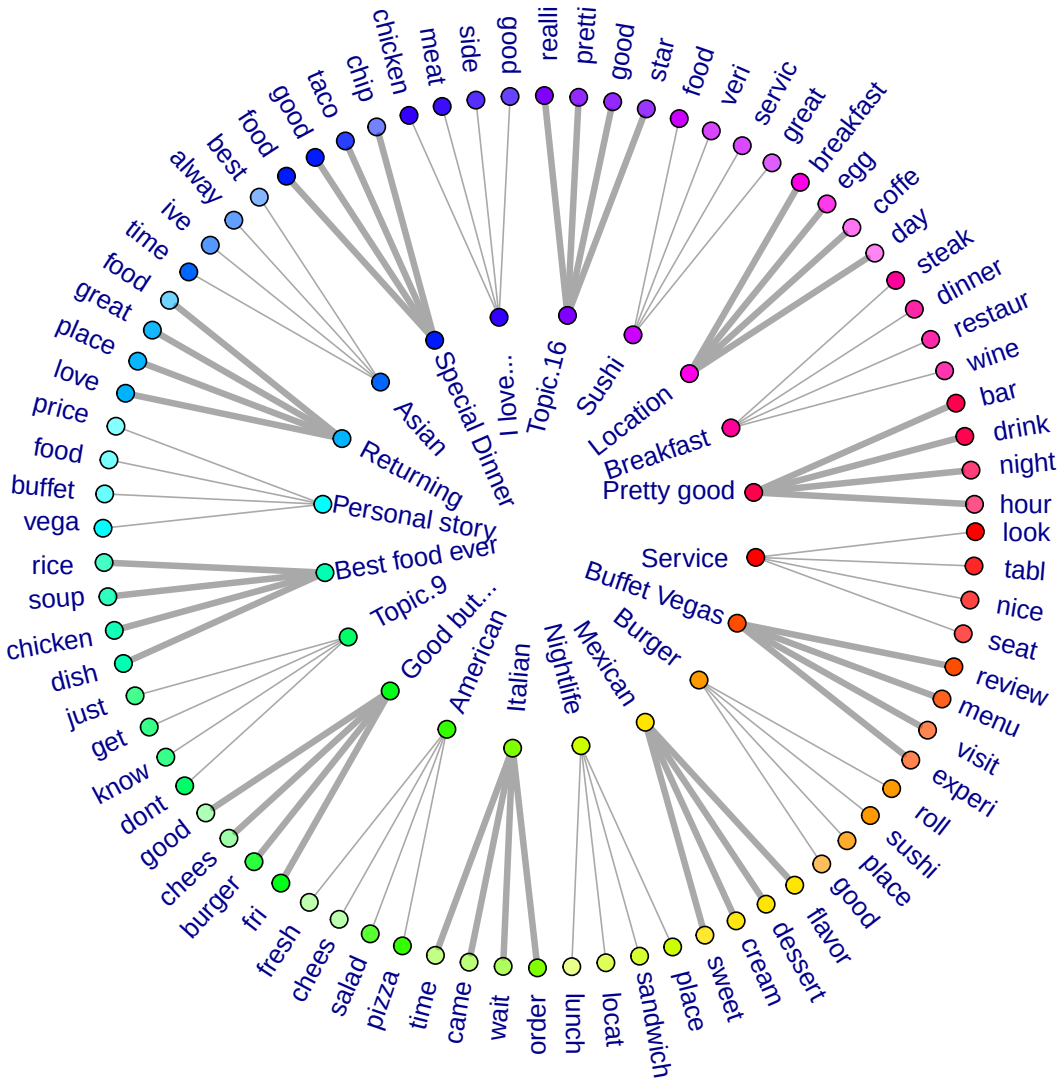

Figure 4: Figure 4. A topic model for positive Yelp restaurant reviews (stars >3). Only the 4 most frequent words are shown for each topic. Color luminance shows the relative in-topic relevance of each term, being the most bright the most frequent term.

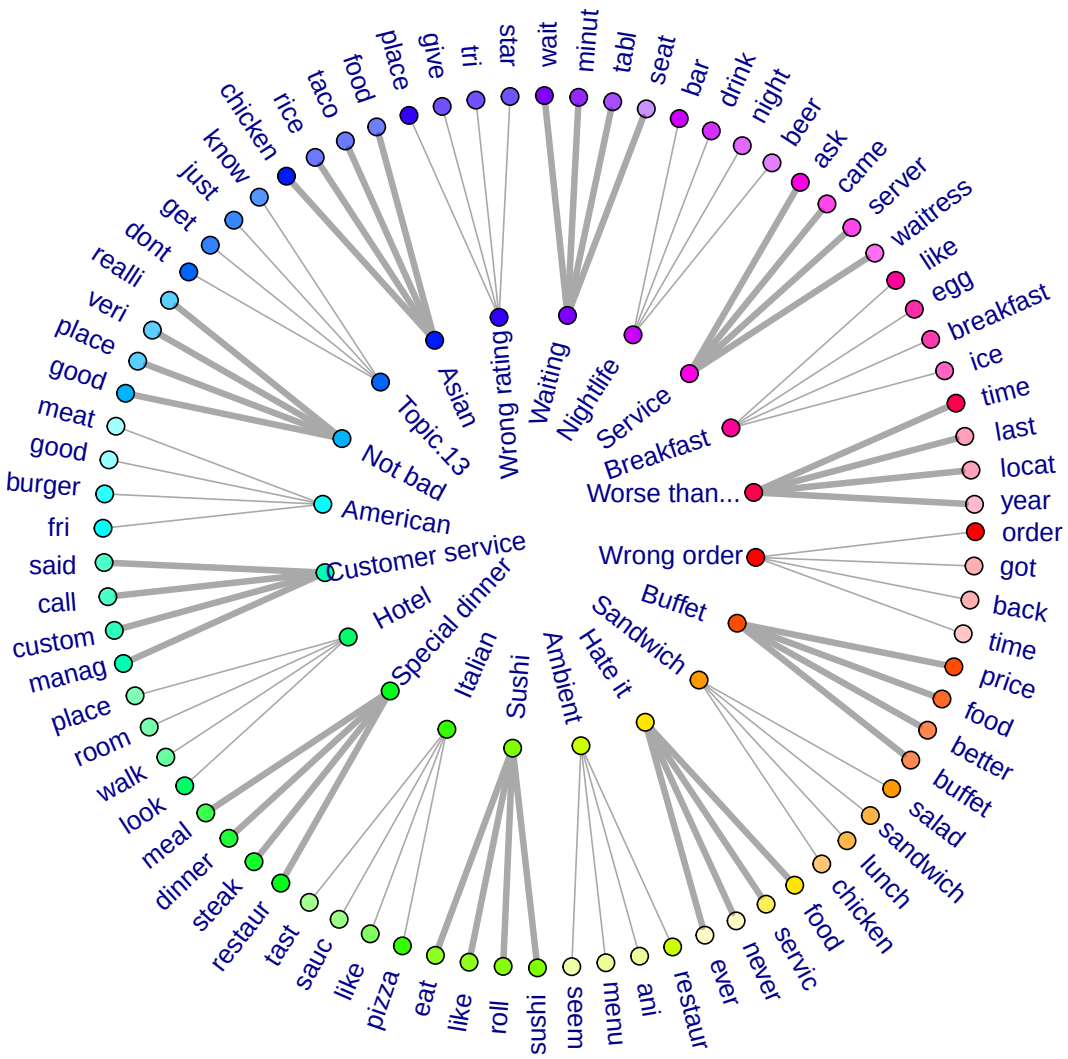

Figure 5: Figure 5. A topic model for negative Yelp restaurant reviews (stars <3). Only the 4 most frequent words are shown for each topic. Color luminance shows the relative in-topic relevance of each term, being the most bright the most frequent term.

Figure 6: Figure 6. Word cloud displaying relevance of different cuisine categories.

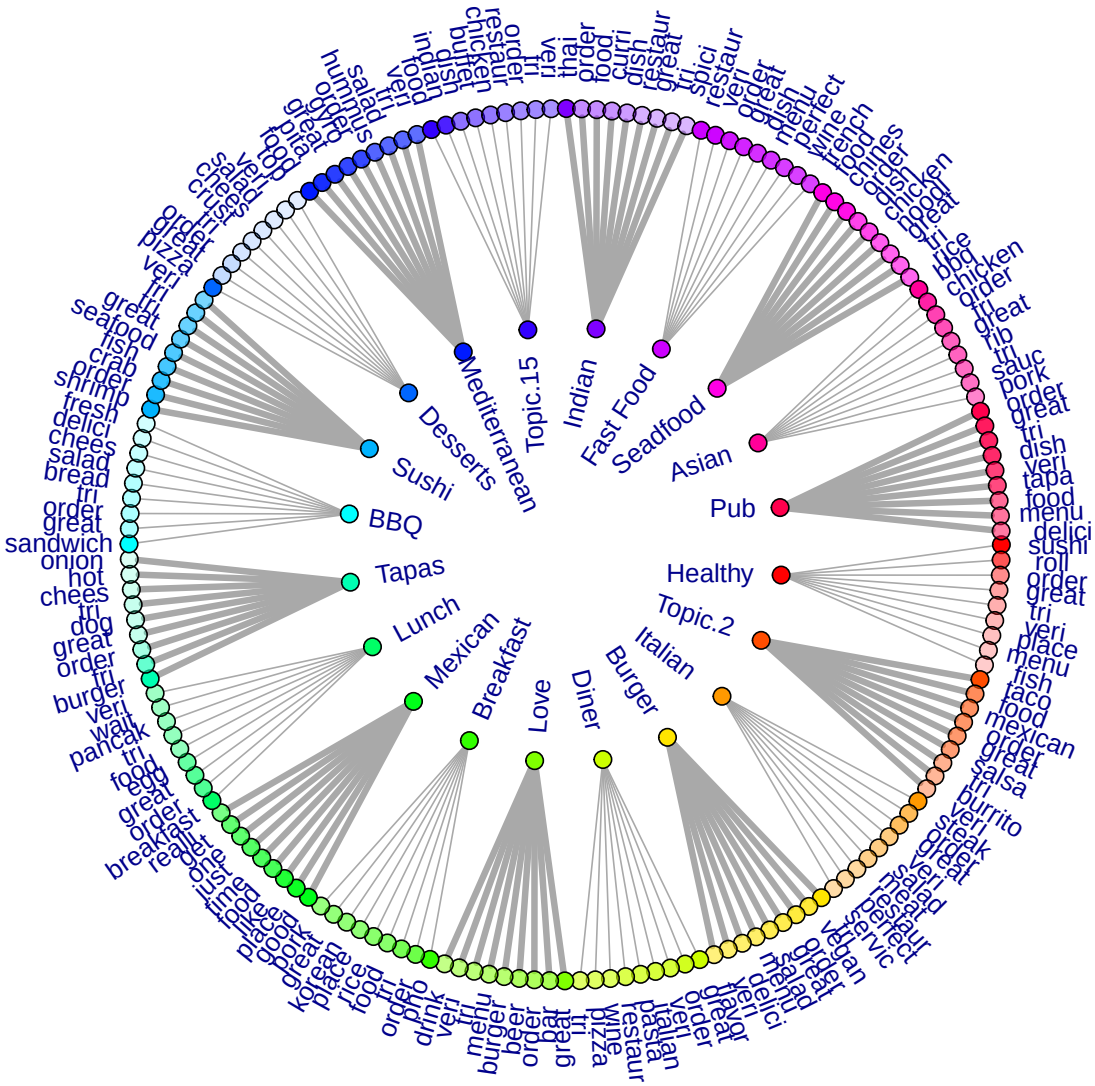

Figure 7: Figure 7. A topic model for positive (stars >3) Yelp restaurant reviews aggregated by cuisine. Only the 9 most frequent words are shown for each topic. Color luminance shows the relative in-topic relevance of each term, being the most bright the most frequent term.

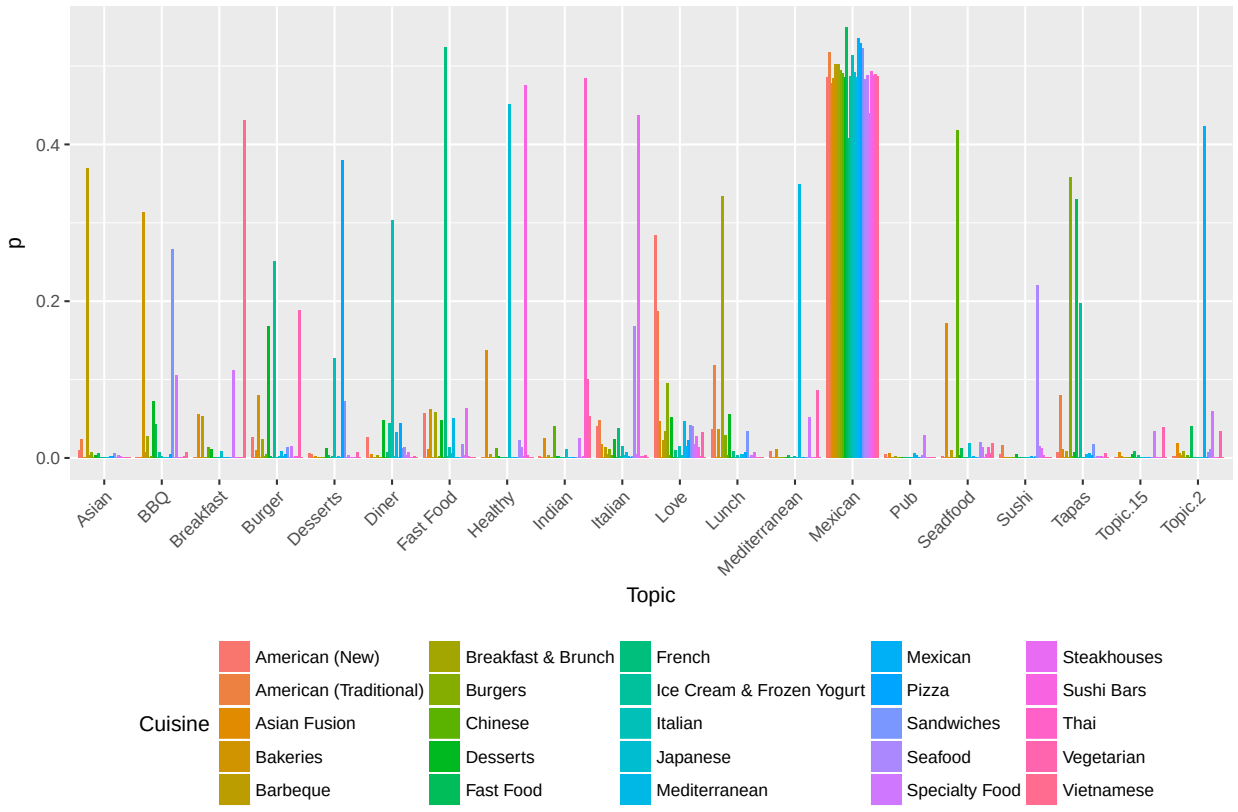

Figure 8: Figure 8. Probability distribution of cuisines over topics. We only show the 25 top reviewed cuisines.

4 Discussion

I have explored some latent topics in a corpus of Yelp reviews for restaurants. For that, I have fitted to the corpus a topic model using LDA with Gibbs sampling.

The topics found display themes related to different cuisines and customer experience. But not all the business labeled as restaturants are a restaurant as show two fo the topics found.

I have further explored the customer experience topics by splitting the corpus in two corpora, one for positive experiences and another for negative experiences and fitting a topic model to each corpus. Many themes that appear in the overall topic model also appear in the new topic models. The new models also show a finer grain decomposition of the customer experience.

The cuisine types found correspond to the most frequent cuisine types listed in the business data. Also, I have aggregated the reviews by cuisine according to the business labels and kept only positive reviews. A new topic model fitted to this corpus focus more on particular aspects of cuisines and aggregates most of the user experience.

5 References

[1] M. Anderson and J. Magruder. "Learning from the Crowd." The Economic Journal. 5 October,2011.

[2] Blei DM, Ng AY, Jordan MI (2003b). "Latent Dirichlet Allocation." Journal of Machine Learning Research, 3, 993–1022.

[3] Ingo Feinerer, Kurt Hornik, and David Meyer (2008). Text Mining Infrastructure in R. Journal of Statistical Software 25(5): 1-54

[4] Kurt Hornik, Christian Buchta, Achim Zeileis (2009) Open-Source Machine Learning: R Meets Weka. Computational Statistics, 24(2), 225-232. doi:10.1007/s00180-008-0119-7

[5] Griffiths TL, Steyvers M (2004). "Finding Scientific Topics." Proceedings of the National Academy of Sciences of the United States of America, 101, 5228–5235.

[6] Phan XH, Nguyen LM, Horiguchi S (2008). "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections." In Proceedings of the 17th International World Wide Web Conference (WWW 2008), pp. 91–100. Beijing, China.

[7] Blei DM, Lafferty JD (2007). "A Correlated Topic Model of Science." The Annals of Applied Statistics, 1(1), 17–35.

[8] Hornik, K., Grün, B., 2011. topicmodels: An R package for fitting topic models. Journal of Statistical Software 40, 1–30.